Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Žiga Emeršič

Razpoznava oseb na podlagi biometričnih podatkov uhljev

magistrsko delo

študijski program druge stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Peter Peer
Somentor: doc. dr. Vitomir Štruc

Ljubljana, 2015

Izjava o avtorstvu magistrskega dela

Spodaj podpisani Žiga Emeršič sem avtor magistrskega dela z naslovom:

Razpoznava oseb na podlagi biometričnih podatkov uhljev

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Petra Peera in somentorstvom doc. dr. Vitomirja Štruca,
- so elektronska oblika magistrskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko magistrskega dela,
- soglašam z javno objavo elektronske oblike magistrskega dela v zbirki "Dela FRI".

V Ljubljani, 15. septembra 2015 Podpis avtorja:

Zahvaljujem se mentorju izr. prof. dr. Petru Peeru za pomoč, veliko zelo koristnih nasvetov in predlogov ter veliko mero potrpežljivosti. Zaslužen je tudi za nastanek dveh konferenčnih objav na temo podatkovne baze in orodja, ki sta del tega magistrskega dela. Zahvaljujem se somentorju doc. dr. Vitomirju Štrucu za zelo koristne nasvete in predloge ter za začetno idejo, brez katere delo sploh ne bi nastalo. Zahvaljujem se tudi vsem, ki so kakorkoli pripomogli k nastanku tega dela.

Kazalo

1 Uvod **1**

1.1 Biometrija in razpoznava 2

1.2 Postopek razpoznave 2

2 Podatkovna baza uhljev **5**

2.1 Sorodna dela 6

2.2 Podatkovna baza uhljev v nekontroliranem okolju 8

2.2.1 Spletni plazilec 10

2.2.2 Orodje za pripravo slik uhljev 14

3 Orodje za razpoznavo uhljev **19**

3.1 Sorodna dela 20

3.2 Arhitektura in delovanje 21

3.2.1 Priprava parametrov 27

3.2.2 Priprava podatkovne baze uhljev 30

3.2.3 Priprava metode za pridobivanje značilk uhljev 31

3.2.4 Priprava odločitvenega modela 34

3.2.5 Izpis in shranjevanje rezultatov 36

4 Pridobivanje značilk uhljev **37**

4.1 Direktna raba normiranih slik 37

4.2 Histogrami orientiranih gradientov 38

4.3 Skalirno invariantna tranformacija značilk 39

4.4 Gosta skalirno invariantna tranformacija značilk 40

4.5 Maksimalno stabilni ekstremi regij 40

4.6 Kvantizacije lokalne faze 40

4.7 Rotacijsko invariantna metoda kvantizacije lokalne faze 41
4.8 Fuzija 41

5 Uporabljeni odločitveni modeli **43**
5.1 Metoda podpornih vektorjev 43
5.2 Metoda k-najbližjih sosedov 44

6 Rezultati **47**

7 Zaključek **53**

Literatura **55**

Seznam uporabljenih kratic

BSIF – Binarized Statistical Image Features – binarizirane statistične značilke slik

DSIFT – Dense Scale-Invariant Feature Transform – gosta skalirno invariantna tranformacija značilk

HD – Hamming Distance – Hammingova razdalja

HOG – Histogram of Oriented Gradients – histogram orientiranih gradientov

ICP – Inner Product Classifier – klasifikator skalarnega produkta

JSON – JavaScript Object Notation – JavaScript objektna notacija

kNN – k-Nearest Neighbors – k-najbližjih sosedov

LPIC – Local Principal Independent Components – metoda lokalnih glavnih neodvisnih komponent

LPQ – Local Phase Quantization – kvantizacije lokalne faze

MSER – Maximally Stable Extremal Regions – maksimalno stabilni ekstremi regij

PCA – Principal komponent analysis – metoda glavnih komponent

RILPQ – Rotation Invariant Local Phase Quantization – rotacijsko invariantna kvantizacije lokalne faze

SIFT – Scale-Invariant Feature Transform – skalirno invariantna tranformacija značilk

SVM – Support Vector Machine – metoda podpornih vektorjev

Povzetek

Na področju biometrije uhljev, kljub razvoju v zadnjih letih, ni na voljo nobenega prosto dostopnega orodja ali podatkovne baze, zajete v nekontroliranem okolju, ki bi olajšal primerjavo metod za razpoznavo oseb na podlagi uhljev. V okviru tega dela je bila pripravljena nova baza uhljev, ki je zajeta v nekontroliranem okolju, orodje za razpoznavo uhljev in metoda rotacijsko invariantne kvantizacije lokalne faze, ki na uhljih še ni bila uporabljena, skupaj s fuzijo te metode z vektorji normiranih slik. Rezultati metode so primerljivi, vendar ne boljši od trenutno najboljših, smo pa s fuzijo dosegli boljše rezultate kot s samo metodo rotacijske invariantne kvantizacije lokalne faze. Razvito orodje za razpoznavo uhljev predstavlja korak k standardizaciji na področju razpoznave uhljev, testi na bazi uhljev pa so pokazali, da baza predstavlja večji izziv od obstoječih, a je hkrati primerljiva in je zato primerna za nadaljno rabo.

Abstract

Human recognition based on ear biometric data

In the field of ear biometrics, despite recent developments, there are no freely available tools or databases captured in the wild that would ease the comparison of methods for ear biometric recognition. A new ear database captured in the wild was developed as a part of this thesis as well as a new toolbox for ear recognition. Rotation invariant local phase quantization method was also applied, for the first time in the field of ear biometrics, together with a fusion of this method with vectors of normalized images. Results are comparable but not better than the state-of-the-art, however, with the fusion we have achieved better results than the rotation invariant local phase quantization alone. The developed toolbox for ear recognition presents a new step towards standardization in the field of ear recognition and tests on the ear database have shown that the database presents greater challenge than the existing ones, but it is still comparable, which makes it suitable for further use.

Poglavje 1

Uvod

Področje razpoznave oseb na podlagi biometričnih podatkov je zanimivo in široko uporabljano. S stališča biometričnih modalnosti so uhlji še posebej zanimivi, ker so, glede na ostale modalnosti (obrazi, odtisi in podobno) manj raziskani in tudi bolj odporni na, na primer, obrazno mimiko, sposobnost razlikovanja dvojčkov, staranje oseb [1] in podobno.

Na področju razpoznave oseb uhljev je trenutna težava korektna primerjava obstoječih metod, saj ni na voljo široko uporabljane in dostopne baze uhljev iz nekontroliranega okolja in orodja za razpoznavo uhljev, ki bi omogočala hitrejši in lažji razvoj na tem področju. Priprava takšne baze, orodja in vpeljava metode rotacijsko invariantne kvantizacije lokalne faze, ki na uhljih še ni bila uporabljena, so glavni nameni tega magistrskega dela.

V uvodnem poglavju so predstavljeni osnovni koncepti biometrije pri razpoznavi oseb. V poglavju 2 so obravnavane podatkovne baze uhljev in predstavljena lastna baza uhljev v nekontroliranem okolju. V poglavju 3 je predstavljeno orodje za razpoznavo uhljev. V poglavju 4 so predstavljene metode za pridobivanje značilk uhljev. V poglavju 5 so predstavljeni odločitveni modeli, ki so bili uporabljeni pri evalvaciji. Vse metode so del orodja za razpoznavo uhljev. Delo se zaključi z rezultati v poglavju 6 in sklepi v poglavju 7.

1.1 Biometrija in razpoznava

Razpoznava oseb na podlagi biometričnih podatkov se v človeških možganih odvija vsakodnevno. V medčloveški komunikaciji ne potrebujemo konstantnih preverjanj osebnih izkaznic, serijskih številk ali osebnih gesel. Medčloveški odnosi temeljijo na stalnem avtomatiziranem medsebojnem razpoznavanju na podlagi biometričnih podatkov: potez obraza, načina hoje, glasu, načina govora. Razlog za prehod in uporabo takih metod v svet računalnikov je tako očiten.

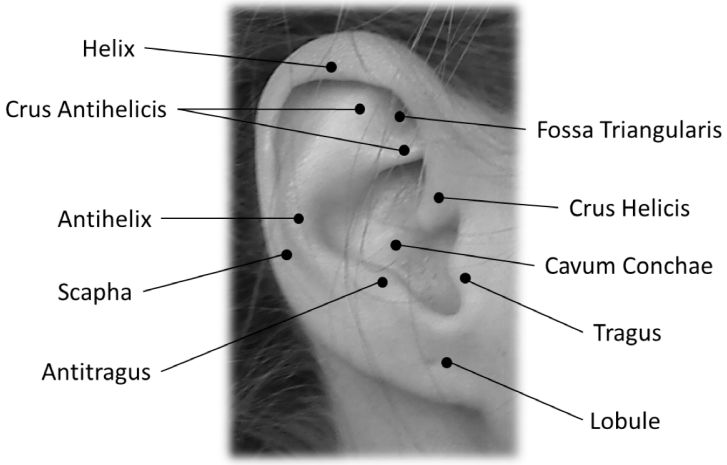
Pri biometriji v računalništvu pogosto razmišljamo le v domeni zagotavljanja varnosti – lažja in za uporabnika preprostejša prepoznava na področjih z omejenim dostopom. Pa vendar domena biometrije v računalništvu ni nujno le dodatek k obstoječim standardnim metodam razpoznave, temveč lahko prinese veliko tudi k uporabniškim vmesnikom, v sistemih, kjer popolna zanesljivost ni nujno potrebna in prikaz informacij namenjenim drugim ni tako ključnega pomena. Računalniški pomočnik, ki prikaže relevantne informacije prilagojene vsakemu uporabniku posebej vsakič, ko se tak uporabnik le približa, je veliko bolj intuitiven kot sistem, kjer je potrebno vedno izbirati uporabnika ali celo vnašati geslo.

Pri razpoznavi oseb lahko uporabljamo več različnih biometričnih podatkov. Metode za pridobivanje delimo na invazivne in ne-invazivne. Pri pridobivanju biometričnih podatkov prstnih odtisov in na primer šarenice gre za invazivne tehnike, saj zahtevajo sodelovanje uporabnika. Za razliko od invazivnih tehnik, pri ne-invazivnih ne potrebujemo direktnega sodelovanja uporabnika. Fotografijo obraza, uhljev, posnetek načina hoje ipd. lahko dobimo brez direktnega sodelovanja in tako ponujajo širši in prijaznejši spekter uporabe. V splošnem so ne-invazivne metode zato bolj zaželene.

Razpoznava oseb na podlagi biometričnih podatkov uhljev predstavlja pomembno modalnost, tako samostojno kot dopolnilno, recimo v kombinaciji z na primer razpoznavo na podlagi obrazov.

1.2 Postopek razpoznave

Postopek razpoznave oseb na podlagi biometričnih podatkov uhljev temelji na zajemu vhodnih podatkov. V splošnem je osnovni postopek razpoznave enak ali vsaj podoben za vse biometrične modalnosti. Vhodni podatki so tako lahko barvne



Slika 1.1: Glavne karakteristike uhlja [1]

slike oseb ali posameznih delov, segmentirani podatki o načinu hoje, binarna slika prstnega odtisa, neobdelane značilke obraznih potez in podobno.

Ko imamo na voljo podatke, ki osebe medsebojno nedvoumno ločijo, lahko pričnemo s postopkom pridobivanja značilk. Namen tega koraka je, da odločitvenemu modelu, ki deluje v vlogi ločevanja med osebami, olajšamo delo, oziroma izboljšamo delovanje celotnega sistema.

Pred pridobivanjem značilk je včasih dobro izvesti postopek predprocesiranja. To pomeni, da odstranimo šumne podatke, slike po potrebi skrčimo ali združimo barvne kanale in podobno. Namen predprocesiranja je izboljšati postopek pridobivanja značilk, in tako posledično izboljšati delovanje celotnega sistema.

Pridobivanje značilk je ključni korak razpoznave oseb. Cilj je čim bolje in čim bolj jasno ter nedvoumno opisati dobljene podatke v domeni razpoznave oseb. Pri fotografiji prstnega odtisa tako na primer želimo čim bolje opisati tiste krivulje, ki so pri razlikovanju posameznih oseb najbolj pomembne. Podobno seveda velja za ostale biometrične modalnosti. Na sliki 1.1 so prikazane glavne značilnosti uhlja, ki služijo kot podlaga za pridobivanje značilk.

Po pridobljenih značilkah lahko te še dodatno izboljšamo, na primer odstranimo ponavljajoče oz. linearno odvisne atribute, ali značilke ustrezno skrčimo in jih s tem pripravimo na boljše delovanje v odločitvenem modelu.

Sledi korak klasifikacije, kjer na podlagi značilk naredimo končno razpoznavo

oseb. Postopek klasifikacije je, skupaj s postopkom pridobivanja značilk, izrednega pomena za delovanje celotnega sistema. Sestavljen je iz dveh faz: faze učenja in faze delovanja oziroma testiranja. Pomembno je, da imamo podatke, ki smo jih dobili v procesu pridobivanja značilk ločene. To pomeni, da med testno in učno množico na eno strani ni prevelikih razlik, da je že učna množica dovolj težka (dovolj blizu resničnemu svetu oziroma okolju ciljne uporabe), vendar hkrati vseeno ne vsebuje koreliranosti, ki bi v postopku testiranja vodila v lažno dobre rezultate[1].

[1]Pojav, kjer se klasifikator preveč prilagaja učni množici (angl. overfitting) je večni problem strojnega učenja in se mu moramo izogibati.

Poglavje 2

Podatkovna baza uhljev

Podatkovna baza uhljev je sestavljena iz slik uhljev različnih oseb. Takšno bazo lahko nato uporabimo za razvoj in evalvacijo metod za razpoznavo oseb na podlagi uhljev.

Slike takih podatkovnih baz lahko vsebujejo tesno obrezana območja okoli uhljev, celotne obraze ali celo več oseb. Skupna lastnost je le ta, da so uhlji relativno dobro vidni[2]. Če uhelj ni prevladujoč objekt na sliki (da zavzema približno več kot $60\% - 70\%$ slike), imamo opravka s podatkovno bazo, nad katero je potrebno pred procesom razpoznave oseb na podlagi uhljev narediti detekcijo uhljev. Takojšnja aplikacija metod za pridobivanje značilk na takšnih slikah, bi vodila v klasificiranje okoliških objektov in anomalij, namesto značilnosti uhljev. To pomeni, da uhelj ali uhlje na sliki lociramo, ločimo od preostanka slike in šele nato izvedemo postopek identifikacije ali verifikacije.

Same slike so lahko različnih velikosti, kakovosti, z različnimi stopnjami šuma, prekrivanji, variacijami osvetlitve ipd. Lahko si predstavljamo, da takšne variacije negativno vplivajo na klasifikacijsko točnost sistema za razpoznavo oseb, vendar si hkrati takšnih variacij želimo pri uporabi učne množice. Sistem za razpoznavo, naučen na bazi slik, ki so blizu slikam, na katere naletimo pri končnem delovanju sistema, bo deloval bolje. Če predpostavljamo, da je najbolj realno in hkrati eno bolj zahtevnih okolij resnični svet, si v kakovostni bazi uhljev želimo slike iz

[2]Relativno vidnost je težko natančno opredeliti. Lahko rečemo, da je uhelj relativno dobro viden, ko se da pridobiti dovolj informacij za uspešno delovanje sistema za razpoznavo oseb na podlagi uhljev.

resničnega sveta.

Pri gradnji sistema za razpoznavo oseb je tako zelo pomembna zmožnost neodvisne preverbe in ocenitve delovanja. V veliki meri to dosežemo s pravo bazo, ki je blizu resničnemu okolju in ki je široko uporabljana.

2.1 Sorodna dela

Na voljo so prosto dostopne podatkovne baze uhljev, vendar pa trenutno na področju razpoznave oseb na podlagi uhljev ni na voljo nobene široko dostopne baze, ki bi bila pripravljena v resničnem okolju (angl. in the wild). To pomeni, da raziskovalci pri razvoju in evalvaciji algoritmov za razpoznavo oseb na podlagi uhljev uporabljajo baze, ki vsebujejo določene pristranskosti. To so baze, ki so bile pripravljene v laboratorijskih pogojih – pri enaki osvetlitvi, pod enakim kotom, uhlji v popolni poravnavi, vse slike narejene z enako fotografsko opremo ipd. Pri pripravi nove podatkovne baze so pomembni vsi našteti kriteriji, hkrati mora biti baza dostopna vsem in dovolj zahtevna, da je zanimiva za uporabo pri razvoju novih metod za prepoznavo na podlagi uhljev.

Dodatno težavo predstavlja dejstvo, da ni ene splošno sprejete podatkovne baze uhljev, ampak se jih uporablja več – to oteži medsebojno primerjavo in pogosto zakrije dejansko kakovost uporabljenih metod za razpoznavo oseb na podlagi uhljev. Razlog za uporabo večih je verjetno v že omenjenih vgrajenih pristranskostih, ki se jih znebimo ali s pripravo nove iz resničnega sveta ali z uporabo večih (pri slednjem pristopu to velja vsaj do neke mere).

Nekateri avtorji podatkovnih baz uhljev so skušali posnemati slike iz resničnega sveta z zajemanjem slik pod različnimi koti in različnimi časi zajema, vendar so pri vseh vseeno odsotne lastnosti, ki nastopijo zaradi zelo različne osvetlitve (zunanja, notranja osvetlitev), uporaba različne fotografske opreme, zajem slik uhljev oseb pri različnih starostih ipd.

Nekaj širše uporabljanih in dostopnih obstoječih podatkovnih baz uhljev:

WPUTEDB

Podatkovna baza Univerze West Pommeranian University of Technology [2] vsebuje 3348 slik 421 oseb s približno 4 do 10 slik na osebo. Slike uhljev so od oseb različnih starosti, vsebujejo prekrivanja in so bile zajete pod različno notranjo

osvetlitvijo in koti od približno 70° do 120°. Slike vsebujejo tako leve kot desne uhlje, kar je tudi razvidno iz imen datotek.

IIT Delhi

Podatkovna baza uhljev Indian Institute of Technology Delhi [3] vsebuje dve bazi uhljev: Delhi I in Delhi II. Obe bazi vsebujeta sivinske slike desnih uhljev (normirane slike so horizontalno zrcaljene in izgleda, kot da gre za leve uhlje) brez večjih prekrivanj. Slike uhljev so bile zajete pod spremenljivo notranjo osvetlitvijo. Delhi I vsebuje 493 slik 125 oseb, Delhi II 793 slik 221. Pri obeh znaša število slik na osebo od 3 do 6. Delhi I je na voljo tako v izvirni kot tudi v normirani obliki (enake dimenzije slik, uhlji centrirani, tesno obrezani in poravnani z osema), medtem ko je Delphi II na voljo samo v normirani obliki. Še posebej zanimiva je primerjava normirane in nenormirane različice Delhi I, saj se vidi, do kolikšne mere lahko vpliva na postopek samo proces normiranja.

University of Notre Dame

Baza uhljev Univerze v Notre Dame vsebuje več ločenih podatkovnih baz uhljev s slikami levih uhljev zajetimi pod spremenljivo notranjo osvetlitvijo. Na voljo so baze s 3480 3D in pripadajočimi 2D profilnimi slikami uhljev 952 oseb, s povprečno 3 do 4 slik na osebo. Na voljo je tudi izključno 2D baza uhljev s 464 slikami 114 oseb s povprečno 4 slikami na osebo [1, 4].

UBEAR

Podatkovna baza uhljev UBEAR vsebuje 4429 slik levih in desnih uhljev 126 oseb [1, 5] s povprečno 35 slik na osebo. Baza je zanimiva, ker so se osebe v času zajetja slik tudi premikale – lastnost, ki je z lastno podatkovno bazo uhljev nismo zajeli. Slike so bile zajete pod različnimi osvetlitvami, pod različnimi koti in vsebujejo prekrivanja.

Pri primerjavi metod za razpoznavo oseb na podlagi uhljev smo znotraj orodja za razpoznavo uhljev, opisanega v poglavju 3, uporabili podatkovne baze WPUTEDB [2], IIT Delhi bazo uhljev [3] in lastno razvito bazo uhljev, opisano v poglavju 2.2. Razlog za izbiro WPUTEDB in IIT Delhi je v njunih lastnostih: WPUTEDB vsebuje barvne slike levih in desnih uhljev s prekrivanji in spremenljivimi koti, medtem ko IIT Delhi vsebuje sivinske slike izključno desnih uhljev brez večjih prekrivanj v obliki las, uhanov ipd.

2.2 Podatkovna baza uhljev v nekontroliranem okolju

Naša podatkovna baza uhljev v nekontroliranem okolju (angl. Computer Vision Laboratory Ear Database – CVLEDB) [6] je sestavljena iz 804 slik 16 oseb s približno 19 do 94 slik na osebo. Ker so slike pridobljene s spleta pomeni, da gre za tako imenovano bazo iz resničnega sveta (angl. database in the wild) in da variirajo glede na velikost, kakovost, osvetlitev, kot pod katerim je uhelj in čas nastanka. Vse slike so shranjene v formatu PNG (Portable Network Graphics). Večina slik je barvnih, 11 slik je sivinskih, vse so manjše od 200×200 slikovnih točk, najmanjša 18×27. Na slikah so prisotna prekrivanja: lasje, uhani, slušalke, kot je to prikazano na sliki 2.1, slike so zajete v zelo različnih časovnih obdobjih. V trenutno dostopnih bazah uhljev so te razlike običajno znotraj nekaj tednov ali mesecev. Pri naši bazi uhljev, predvidevamo, da razlike znašajo tudi do 30 let – na podlagi začetnih (torej neobrezanih) fotografij, kjer so bile nekatere osebe fotografirane v različnih življenjskih obdobjih. Povprečna razlika, predvidevamo, znaša nekaj let.

Vse slike vsebujejo obrezane slike uhljev, to pomeni, da ne vsebujejo pripadajočih obrazov, lahko pa vsebujejo bližnjo okolico uhljev. Kljub temu ocenjujemo, da detekcija za doseganje dobrih rezultatov ni potrebna, med drugim tudi zato, ker anotacijski podatki vsebujejo informacije o središču uhlja. Če bi slike vsebovale celotne obraze, bi se bazo lahko uporabljalo tudi za namene evalvacije detekcije uhljev, vendar bi se v tem primeru lahko pojavile težave z dovoljenji oseb za javno objavo. V trenutnem načinu, kjer celotni obrazi oseb niso vidni je zagotovljena anonimnost oseb, saj oseba tako ni prepoznavna, v bazi pa je shranjena le s številskim identifikatorjem. Hkrati je fokus našega dela posvečen samemu pridobivanju značilk in izgradnji pripadajočega orodja, čeprav drži, da bi se z detekcijo zagotovo izboljšala splošna natančnost.

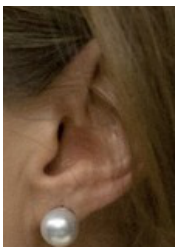
Avtorji hranimo tako izvorne, neobrezane slike kot tudi izvorne URL naslove vseh slik, vendar tega v bazi dostopni javnosti, zaradi anonimizacije oseb ni (čeprav gre za javno izpostavljene osebe). Takšen način hrambe in prikaza slik v podatkovni bazi uhljev je bil usklajen z mnenjem Informacijskega pooblaščenca Republike Slovenije.

(a) Velik kot

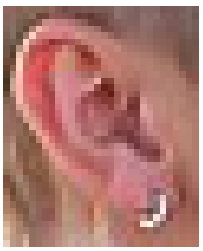
(b) Sivinska slika

(c) Rahlo prekrivanje z lasmi in uhanom

(d) Težje prekrivanje z lasmi



(e) Težje prekrivanje z lasmi in uhanom



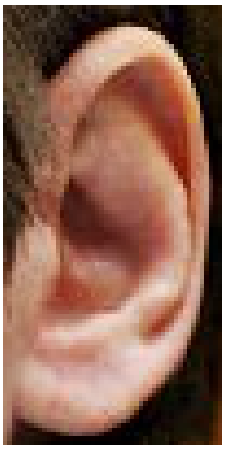
(f) Slika nizke ločljivosti

Slika 2.1: Tipični primeri zajetih slik uhljev v nenadzorovanem okolju

Ker gre za slike iz nenadzorovanega okolja oziroma resničnega sveta, naj bi tako dobro posnemale resnična okolja, v katerih se lahko znajde sistem za razpoznavo oseb na podlagi uhljev.

Slike uhljev vsake osebe so shranjene v podmapah – ime mape določa identifikacijsko številko osebe. Če želimo v orodju opisanem v poglavju 3.4 uporabljati druge baze (na primer IIT Delhi), moramo slike urediti v podmape. Za ta namen smo pripravili PHP skripto, ki glede na imena sortira slike v podmape in je na voljo skupaj z orodjem (kriterij za delitev je del imena datoteke). Znotraj vsake mape osebe se nato nahaja opcijska tekstovna datoteka tipa JSON, ki vsebuje anotacije za vse slike uhljev osebe. Primer takšne anotacije je prikazan na sliki 2.2. Razlog za PHP skripto in ne Bash ali Shell je sistemska neodvisnost kot tudi dejstvo, da tako spletni plazilec, opisan v poglavju 2.2.1 in orodje za pripravo slik uhljev, opisano v poglavju 2.4, tečeta v PHP okolju.

```
[
  "347": {
    "file": "347.png",
    "x": 13,
    "y": 62,
    "d": "l",
    "w": 50,
    "h": 101
  }, ...
]
```

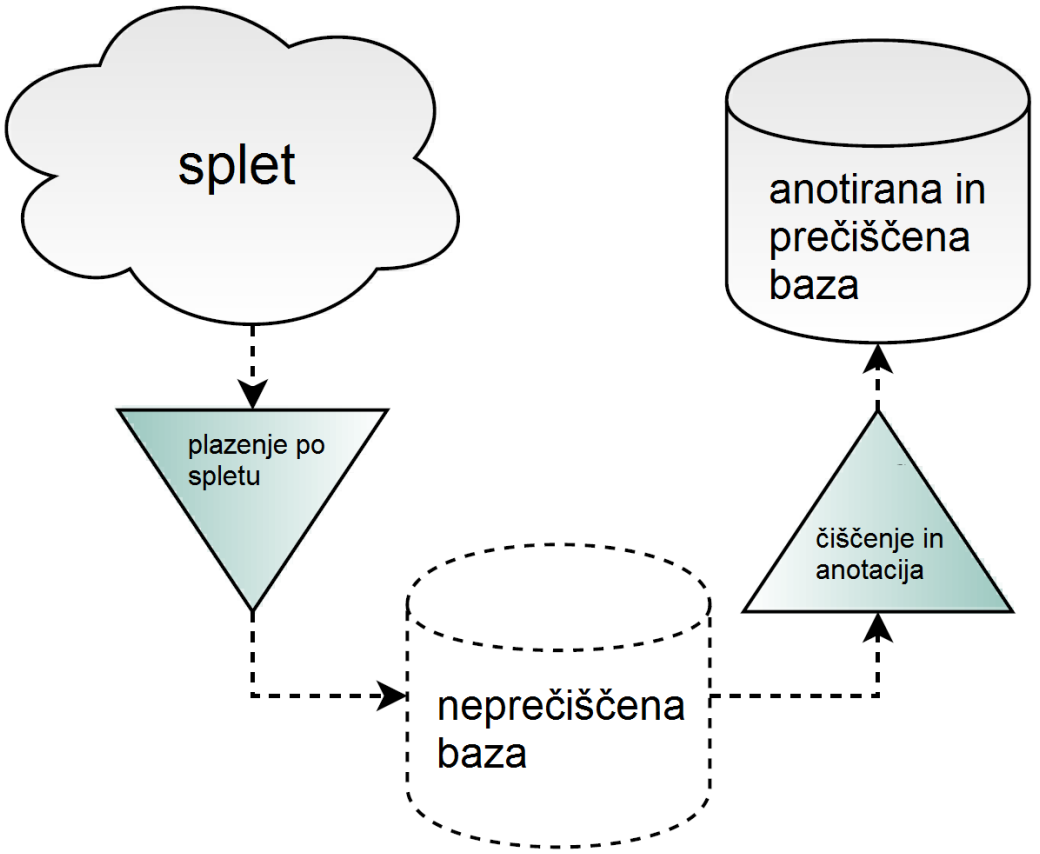


Slika 2.2: Primer JSON anotacije posamezne slike uhlja s pripadajočo sliko. Atribut `file` hrani ime datoteke uhlja, ki ga opisuje anotacija, atributa `x` in `y` predstavljata koordinate sredine tragusa, `d` usmerjenost – `l` = levo (angl. left), `r` = desno (angl. right), atributa `w` in `h` širino (angl. width) in višino (angl. height). Anotacija hrani tudi identifikacijsko številko osebe, vendar je podatek posredno na voljo tudi preko imena mape, v kateri se slika uhlja nahaja

Za namene pridobivanja slik uhljev sta bili razviti dve orodji: spletni plazilec (angl. web crawler) in orodje za pripravo slik uhljev. S spletnim plazilcem samodejno pridobimo slike s spleta [7], ki jih nato z orodjem za pripravo slik uhljev lokalno uredimo in anotiramo ali zavržemo neustrezne. Konkretni postopek pridobivanja slik uhljev je prikazan na sliki 2.3. Obe orodji tečeta na spletnem strežniku Apache, vse datoteke se hranijo na datotečnem sistemu. Skripte so napisane v PHPju. Uporabniški vmesnik je pripravljen v kombinaciji s HTML5 in JavaScriptom, saj omogočata zelo hitro pripravo uporabniškega vmesnika in zagotavljata platformno neodvisnost.

2.2.1 Spletni plazilec

Spletni plazilec se za osnovni nabor slik zanaša na podatke programskih iskalnih vmesnikov Google Custom Search API (Alphabet) [8] in Bing Search API (Mi-



Slika 2.3: Diagram kreiranja naše baze uhljev v nekontroliranem okolju

crosoft) [9]. Pri obeh je potrebna registracija in pridobitev ključa, ki se nato uporablja pri zahtevkih za pridobivanje podatkov. Oba ponujata storitve brezplačno do določenega števila zahtevkov, vendar je potrebna uporaba omenjenega ključa, ki nato služi kot podlaga za morebitno obračunavanje in omejevanje storitev. Pri Google Custom Search APIju je omejitev 100 zahtev na dan in 1 zahteva na sekundo, pri Bing Search APIju 5.000 zahtevkov na mesec (v času nastajanja tega dela 2014 in 2015), brez eksplicitne omejitve hitrosti zahtev. Vsak zahtevek je bil opravljen s korakom 50, to pomeni, da vsak nabor rezultatov vsebuje 50 URL naslovov.

Kot ključne besede za iskanje so bili uporabljeni nizi sestavljeni iz imena znane osebe in besede "side" (na primer "Edward Snowden side"). Z iskanjem znanih oseb si zagotovimo dovolj veliko število slik. Enako velja za ključno besedo "side",

saj je bilo empirično ugotovljeno, da daje boljše rezultate kot na primer "Edward Snowden ears" ali "Edward Snowden side face".

Oba vmesnika ponujata dva načina vračanja podatkov:

1. *tekstovno iskanje* – vmesnik vrne seznam URL naslovov spletnih strani, kjer se nahajajo podatki o iskanem nizu,
2. *slikovno iskanje* – vmesnik vrne seznam URL naslovov slik, ki po kriterijih iskalnika ustrezajo iskanemu nizu.

Glede na vrsti pridobljenih podatkov sta nato načina za pridobivanje slik uhljev različna. Pri slikovnem iskanju se slike, ki se nahajajo na ciljnem URLju, shranijo direktno, pri tekstovnem iskanju pa je potrebno rekurzivno preiskovanje slikovnih elementov v HTML DOM dokumentih do globine 3. Za oba načina so bile razvite PHP skripte, ki kličejo ustrezne spletne iskalne vmesnike, analizirajo vsebino in shranijo podatke uhljev. Uporabniški vmesnik je podprt z JavaScriptom in Ajaxom zaradi zagotavljanja boljših povratnih informacij med izvajanjem pridobivanja slik uhljev. V naslednjih dveh podpoglavjih sta oba postopka podrobneje opisana.

Začetni cilj pridobivanja slik je bil pridobiti 500 slik za vsak iskalni niz (to je 500 na osebo) in nato zavreči neustrezne slike, ustrezne pa anotirati in izrezati uhlje. Ugotovili smo, da pri takšnem številu zahtevanih slik in pri ključnih besedah tipa "ime priimek side" povsem zadošča zgolj slikovno iskanje. Tekstovno iskanje se izkaže za uporabno, če iščemo bolj splošno, na primer "ear". V primeru iskanja s splošno ključno besedo "ear" se zgodi, da postajajo slike pri slikovnem iskanju pri vsakem naslednjem zahtevku (uporabili smo korak 50) slabše in bolj oddaljene od začetnega iskalnega niza že po štirih iteracijah – število slik, ki jih lahko pridobimo se izčrpa. Le v takih primerih postane spletno iskanje v kombinaciji s plazenjem po dokumentih uporabno.

Tekstovno iskanje

Po začetni pridobitvi seznama URL naslovov spletnih strani, se prične rekurzivni postopek obiska strani, analize in morebitnega nadaljnjega obiska strani na URL naslovih prisotnih v preiskanem dokumentu. Vhod v spodaj opisani postopek predstavlja URL naslov, izhod vsake veje rekurzivnega postopka je shranjena slika uhlja ali prazen rezultat:

1. Obisk in pridobitev glave: Na podlagi URLja se preveri dostopnost in pridobi glavo spletnega dokumenta z zahtevo tipa `HEAD`.
2. Analiza glave: Če je v glavi informacija, da gre za slikovni tip, se po pridobivanju telesa dokumenta vsebino shrani direktno na disk kot morebitno sliko uhlja (korak 4). Če se v glavi nahaja informacija, da gre za dokument tipa DOM se izvede analiza dokumenta v koraku 5 (korak 4 se preskoči).
3. Pridobitev dokumenta: Naredi se ponovni zahtevek, tokrat tipa `GET`. S tem pridobimo celotno vsebino na trenutnem URL naslovu. Razlog, da se pridobivata glava in celoten dokument ločeno[3], je v velikih dokumentih nepodprtih formatov – na primer dokument tipa PDF velikosti reda nekaj 10 MB. Nepotrebno pridobivanje takšnega dokumenta povzroči veliko obremenitev in izredno upočasni ali celo onemogoči pridobivanje podatkov (izmerjene upočasnitve so bile reda velikosti nekaj 10 minut kljub optični mrežni povezavi). Dvojni zahtevi `HEAD` in `GET` tako vseeno predstavljata manjše izgube, saj so meritve pokazale izgube reda velikost 100 milisekund na zahtevek, kar tudi ob velikem številu zahtevkov ne predstavlja pomembnih upočasnitev.
4. Morebitno shranjevanje: Če se v koraku analize glave (korak 2) ugotovi, da so podatki slikovnega tipa, se na tem mestu naredi shranjevanje na disk, trenutna veja rekurzivnega postopka pa se konča.
5. Analiza dokumenta: Analiza se izvede, če je bilo ugotovljeno, da gre za DOM dokument. Postopek poteka tako, da se pregleda vse HTML `a` (angl. anchor) in `img` (angl. image) elemente. Atribut `href` pri elementih tipa `a` služi kot vhod v novo rekurzivno vejo (korak 1), če ni bila presežena globina 3 in če povezava ali napis pri povezavi vsebuje ključno besedo "ear". Takšne omejitve po ključni besedi so potrebne, saj so poskusi pokazali, da se drugače obišče preveč povezav, ki so nepovezane z začetnim iskalnim nizom. Podobne omejitve so uporabljene pri slikah (`img`), kjer se upoštevata atributa `src` za povezavo in `alt` kot opis, pri katerem se preverja ključno besedo. Obe vrsti povezav (`a` in `img`) se obravnavata enako in služita kot vhod v novo rekurzivno vejo (korak 1). Razlog za to je ta, da se vedno, torej tudi pri navidez veljavnih slikovnih elementih (`img`), zagotovi ustrezno preverjanje dostopnosti in veljavnosti.

[3]Potrebno se je zavedati, da pri zahtevku tipa `GET` ali `POST` dobimo tudi glavo in ne samo telesa spletnega dokumenta.

Slikovno iskanje

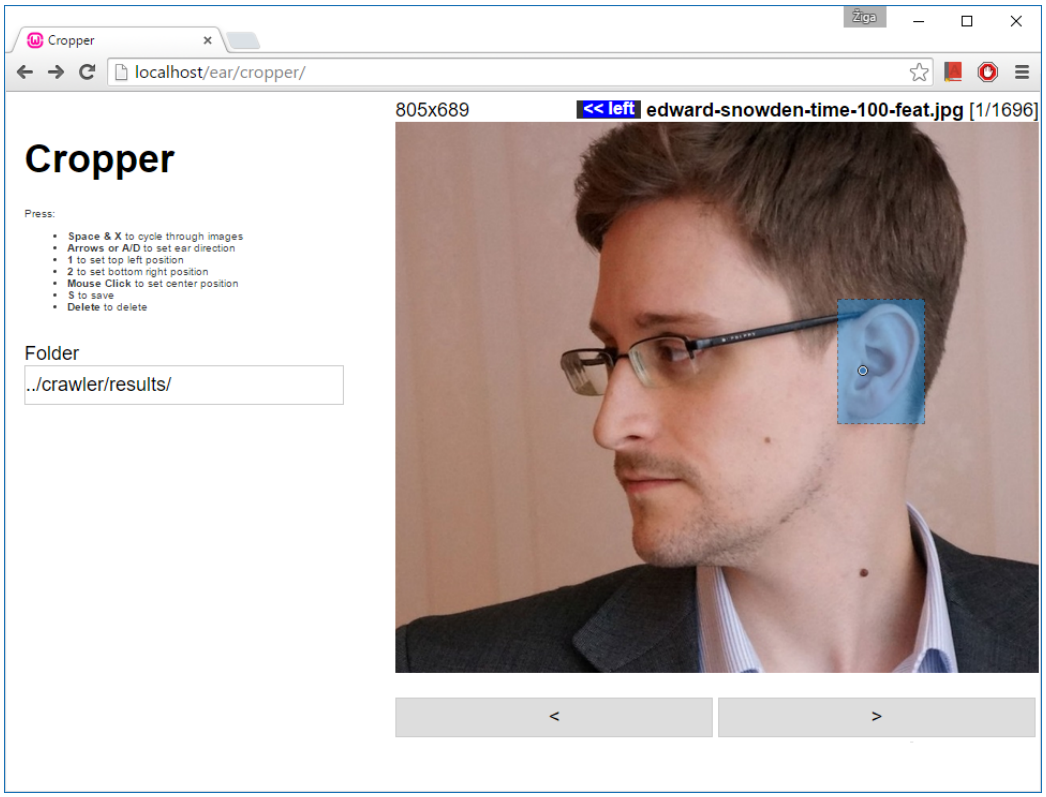
Postopek pridobivanja slik v načinu slikovnega iskanja je preprostejši od plazenja po DOM dokumentih. Tako kot pri tekstovnem iskanju začnemo s seznamom URL naslovov, s to razliko, da tu vnaprej vemo, da gre izključno za slike. Postopek je iterativne narave, kjer se s sprehodom skozi seznam URL naslovov obiskuje ciljne dokumente (slike) in shranjuje slike na disk.

2.2.2 Orodje za pripravo slik uhljev

Po opravljenem postopku opisanem v poglavju 2.2.1 imamo shranjenih veliko slik, med katerimi pa je le majhen delež uporabnih – od 5.000 slik jih je približno 100 (2%) uporabnih. To pomeni, da je potrebno slike izločiti in jih ustrezno anotirati.

Postopek zavračanja neustreznih slik lahko avtomatiziramo, vendar postopek ni trivialen, saj moramo zagotoviti 100% zanesljivost. V realnosti si lahko postopek le nekoliko poenostavimo. Potrebno se je tudi zavedati, da je že zgrajeno podatkovno bazo slik uhljev naknadno preveriti veliko težje kot na primer bazo slik obrazov, saj ljudje razlikujemo osebe po obrazih brez večjih težav, medtem ko za uhlje to ne velja. Avtorji dela [7] pri pridobivanju slik s spleta priporočajo rabo tekstovnega iskanja v kombinaciji s kasnejšim filtriranjem s pomočjo SVM klasifikatorja. S tem so dosegli boljše rezultate od slikovnega iskanja, vendar je potrebno poudariti, da so avtorji klasificirali zgolj vrsto objektov na slikah (recimo pingvin ali ne pingvin) in tudi niso potrebovali popolnoma 100% zanesljivosti. Pri gradnji podatkovne baze uhljev klasificiranih po osebah se je potrebno zavedati, da ni dovolj, da ločimo fotografije uhljev oziroma profilnih fotografij oseb, temveč tudi ali konkretna fotografija uhlja pripada iskani osebi ali komu drugemu. S praktičnega stališča si tako z rabo dodatnega klasifikatorja lahko le nekoliko olajšamo delo, povsem avtomatizirati postopka nastajanja podatkovne baze uhljev pa vsaj trenutno ni mogoče.

Omenimo naj tudi, da trenutno (2014, 2015) dajeta tako Google Custom Search API (Alphabet) [8] kot tudi Bing Search API (Microsoft) [9] pri konkretnem iskanju tipa "ime priimek side" rezultate, ki vsebujejo malo slik, ki niso povezane z iskalnim nizom. To pomeni, da bi morali vložiti kar nekaj truda v klasifikator, ki bi pomagal pri izločanju slik. Zaradi tega se na delu pridobivanja slik nismo odločili za rabo



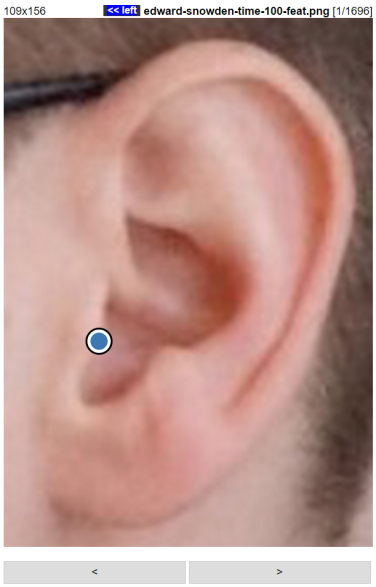
Slika 2.4: Orodje za izločanje in anotacijo slik

klasifikatorjev, temveč smo uporabili izključno ročno izločanje in anotiranje.

Zaradi omenjenih razlogov in dejstva, da je potrebna 100% natančnost, smo se odločili za ročni pristop, kjer se s pomočjo orodja izloči neveljavne slike. Med neveljavne slike štejemo tiste s prenizko ločljivostjo, prevelikim prekrivanjem, prevelikim kotom ali z objekti, ki niso iskana oseba in njeni uhlji. Stroge meje kdaj ima slika prenizko ločljivost ni, slike so se izločale glede na ostale pridobljene slike. Potrebno se je tudi zavedati, da ima lahko slika veliko ločljivost, vendar je lahko zamegljena ali kako drugače popačena.

Zaradi potrebne natančnosti smo izreze uhljev naredili ročno. Avtomatizem namreč ne zagotavlja 100% zanesljivosti, hkrati pa zahteva čas za razvoj in implementacijo.

Orodje za takšno izločanje in anotiranje slik uhljev je prikazano na sliki 2.4. Orodje omogoča hitrejše obrezovanje, kot če bi to počeli ročno v kakšnem ob-



Slika 2.5: Rezultat obrezave in anotacije

stoječem urejevalniku slik. Za vir slik služi mapa s shranjenimi slikami, ki smo jih pridobili z orodjem opisanem v poglavju 2.2.1. Med slikami se premikamo s pomočjo puščic na uporabniškem vmesniku ali tipkama `presledek` ali `x`. S tipko `1` določimo zgornji levi kot, s tipko `2` spodnji desni. Obe koordinati se določita glede na trenutni položaj kurzorja miške. S klikom na levi miškin gumb določimo sredinsko točko zunanjega dela tragusa (na sliki 2.4 označeno z modrim krogom). S puščičnimi tipkami določimo smer uhlja. Poudariti je potrebno, da ne gre nujno za levi ali desni uhelj, saj smo opazili, da so slike na spletu velikokrat zrcaljene – to je najbolj opazno v primerih, ko so za osebo določeni napisi pri katerih je zrcaljenje očitno. Smer uhlja tako ne pomaga nujno v procesu klasifikacije (kjer bi ločevali leve in desne uhlje), ampak pri procesu pridobivanja značilk, kjer je pri določenih algoritmih lahko pomembno, v katero smer se širijo grbine uhlja.

Po ustrezni anotaciji slike se shrani samo obrezano območje, pripadajoči podatki (sredinska točka, smer uhlja, dimenzije slike) se shranijo v pripadajočo datoteko `annotations.json`. Rezultat, ki sicer ni del naše baze uhljev je prikazan na sliki 2.5. Na sliki je vidno, da mesto tragusa ni označeno najbolje, uhelj ni tesno obrezan – prisoten je del očal, velik del slike zavzemajo lasje in koža ob uhlju.

Vse omenjene akcije se izvajajo v brskalniku, rezultati pa se pošljejo na strežnik, kjer se obrezana slika, skupaj z anotacijsko datoteko, shrani na datotečni sistem strežnika.

Poglavje 3

Orodje za razpoznavo uhljev

Naše orodje za razpoznavo uhljev (angl. Computer Vision Laboratory Ear Toolbox – CVLET) [10] je bilo razvito z namenom poenostaviti standardne postopke, ki jih morajo vedno znova opravljati raziskovalci, ko se lotijo izdelave novih metod (ali preverbo obstoječih) za razpoznavo oseb na podlagi biometričnih podatkov uhljev. Zgolj razvoj metode za pridobivanje značilk uhlja ni dovolj. Vedno je potrebno pridobiti bazo uhljev, pripraviti programsko kodo za branje, pripraviti ustrezno podporno kodo za klicanje metod. Ko imamo pridobljene značilke, ki dobro opisujejo uhlje, je potrebno pripraviti še ustrezne odločitvene modele, jih naučiti in sprogramirati še ta del podporne programske kode. Sledi še del priprave kode za prikaz in interpretacijo rezultatov.

Vse to po nepotrebnem vzame veliko dragocenega časa, ki bi se lahko porabil za razvoj novih, potencialno boljših metod za pridobivanje značilk uhljev. Še več – če vsak raziskovalec razvija te dele sam, se po nepotrebnem vpeljuje možnost napak in variacij, ki lahko pomembno vplivajo na končne rezultate. To tudi oteži korektno primerjavo razvitih algoritmov za pridobivanje značilk uhljev.

Pri celotnem orodju za razpoznavo uhljev je poudarek na razvoju novih metod za pridobivanje značilk uhljev, ki jih čim bolje opišejo in omogočijo čim boljšo razpoznavo oseb na podlagi teh značilk. Tudi v celotnem magistrskem delu je tako večji poudarek na metodah za pridobivanje značilk uhljev kot na odločitvenih modelih. Razvoj odločitvenih modelov je namreč lahko povsem ločena veja, ki spada pod strojno učenje, naš fokus pa je na metodah računalniškega vida.

Samo orodje omogoča izvajanje kombinacij metod za pridobivanje značilk in

odločitvenih modelov. Pri določenih vektorjih (torej zapisih značilk uhljev) se lahko določeni odločitveni modeli obnašajo povsem drugače od drugih in je zato pomembno, da je možnost kombiniranja omogočena, hkrati pa je smiselno, da nista procesa odločanja in pridobivanja značilk združena, saj lahko na ta način isti odločitveni model uporabljamo na večih različnih metodah za pridobivanje značilk uhljev in obratno.

Ločenost odločitvenega dela od dela pridobivanja značilk je smiselna tudi zato, da se raziskovalci lahko osredotočijo le na slednjega. Primerjava med različnimi metodami pa tako poudari razlike med samimi metodami pridobivanja značilk, saj je odločitveni model med različnimi avtorji lahko enak (ali obratno). Na ta način se ne more zgoditi, da bi bila določena metoda pridobivanja značilk izpostavljena kot izrazito dobra ali slaba samo zato, ker so avtorji uporabili zelo dober ali zelo slab odločitveni model, ki pa s samimi uhlji nima nujno direktne povezave.

Orodje je razvito v Matlabu in je prosto dostopno.

3.1 Sorodna dela

Za evalvacijo metod za razpoznavo oseb na podlagi biometričnih podatkov uhljev v času pisanja ni na voljo nobenega odprto dostopnega orodja. Na voljo je nekaj splošnih [11, 12] in specifičnih, namenjenih za obraze [13, 14].

Za splošno rabo na področju biometrije je zanimiva platforma za evalvacijo in testiranje biometričnih modalnosti (angl. Biometrics Evaluation and Testing – BEAT) [12]. Ponuja dober in zmogljiv vmesnik, širok nabor orodij za eksperimente na področju biometrije. V okviru platforme je na voljo tudi dinamično odmerjanje računskih virov in zakup le-teh. Ponuja dobro ponovljivost eksperimentov in omogoča rabo tudi v pedagoške namene. Glavne pomanjkljivosti, ki jih naše orodje odpravlja, je dostopnost izključno preko spleta (preko spletnega vmesnika), raba interne programske kode in odsotnost integrirane baze uhljev iz resničnega sveta, ki je del orodja za razpoznavo uhljev.

Sicer pa je bil problem primerjave rezultatov v biometriji že velikokrat izpostavljen [15–17].

Pomemben aspekt orodja za evalvacijo je poleg zmogljivosti, transparentnosti in omogočanja primerljivosti rezultatov tudi enostavnost uporabe [18].

3.2 Arhitektura in delovanje

Orodje za razpoznavo uhljev je sestavljeno iz dveh glavnih inicializacijskih datotek (`cvlet_init_and_run.m` in `cvlet_run.m`), vhodne (konfiguracijske) datoteke (`START_HERE.m` – poimenovana z velikimi črkami, da jo uporabnik takoj vidi), podpornih – jedrnih funkcij (znotraj mape `core`) in funkcij za pridobivanje značilk (znotraj mape `extractors`) ter odločitvene modele (znotraj mape `models`).

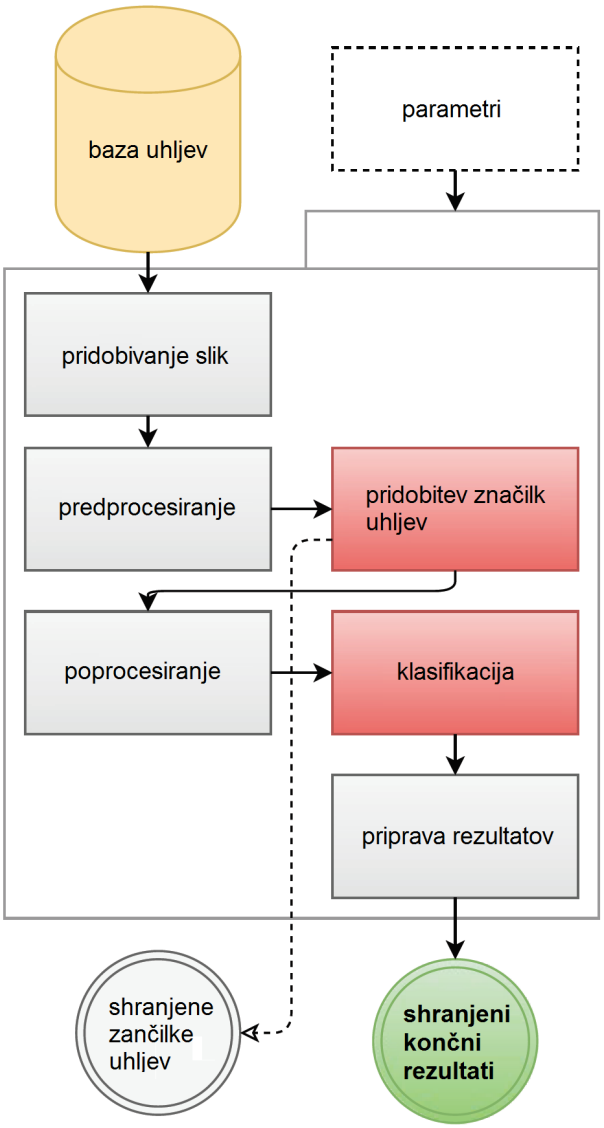
Osnovna konfiguracija je pripravljena tako, da lahko uporabnik prične z uporabo brez kakršnegakoli spreminjanja oziroma prilagajanja. Orodju je že priložena naša podatkovna baza uhljev, opisana v poglavju 2.2, seveda pa lahko uporabnik uporabi tudi drugo bazo. Osnovno delovanje orodja je prikazano na sliki 3.1.

Pri testih izvedenih v okviru tega magistrskega dela so bile uporabljene tudi baze, ki pa niso priložene orodju za razpoznavo uhljev: IIT Delhi I, IIT Delhi II in WPUTEDB. Razlogi so v licencah, ki ne dovoljujejo prostega širjenja slik, saj so baze na voljo ob izrecni zahtevi za dostop, prosto širjenje ni dovoljeno.

Na sliki 3.2 je prikazano glavno delovanje orodja za razpoznavo uhljev v obliki poenostavljene Matlab kode.

```
function cvlet_run(database, extractor, model)
  % PREDNALAGANJE
  [status, data] = cvletcore_preload(database, extractor);

  if (isequal(status, 1))
    % UPORABA PREDNALOZENIH ZNACILK
    features = data;
  else
    % BRANJE BAZE UHLJEV
    if (isequal(status, 2))
      % UPORABA PREDNALOZENE BAZE
      db = data.db;
      annotations = data.annotations;
    else
      % NOVO BRANJE BAZE
      [db, annotations] = cvletcore_database_load(database.path);
```



Slika 3.1: Diagram delovanja razvitega orodja za razpoznavo uhljev

```
      % SHRANJEVANJE BAZE za kasnejso rabo
      cvletcore_store(2, struct('db', db, 'annotations', ...
        annotations), database, extractor);
    end

    % OPCIJSKA INICIALIZACIJA metode za pridobivanje znacilk
    cvlet_init_extractor();

    % OPCIJSKO PREDPROCESIRANJE
    [db, annotations] = cvletcore_preprocess_all(db, annotations);

    % PRIDOBIVANJE ZNACILK
    features = cvletcore_features_extract_all(db, annotations);

    % OPCIJSKO POSTPROCESIRANJE
    features = cvletcore_postprocess_all(features);

    % SHRANEVANJE ZNACILK
    cvletcore_store(1, features, database, extractor);
  end

  % OPCIJSKA INICIALIZACIJA odlocitvenega modela
  cvlet_init_model();

  % EVALVACIJA
  [targets, outputs] = cvletcore_evaluate(features);

  % PRIKAZ IN SHRANJEVANJE REZULTATOV
  cvletcore_vizualize(targets, outputs);
end
```

Slika 3.2: Poenostavljena Matlab koda delovanja orodja za razpoznavo uhljev

Na sliki 3.3 vidimo osnovno logiko izbire aktivne baze, metode za pridobivanje značilk in odločitvenega modela. S pomočjo Matlab ukaza `addpath` se v okolje doda mapa s programsko kodo trenutno izbranega modela in metode za pridobivanje značilk.

```
function cvlet_init_and_run(databases, extractors, models, ...
  crossvalind, log_level)

    % ZACETNA INICIALIZACIJA
    global_init()

    % SPREHOD SKOZI VSE KOMBINACIJE baz, metod za pridobivanje
    % znacilk in modelov ter klic glavne veje izvajanja
    % cvlet_run(...)
    for id = 1:numel(cvlet.databases)
        for ie = 1:numel(cvlet.extractors)
            for im = 1:numel(cvlet.models)

                cvlet.current_database = cvlet.databases(id);
                cvlet.current_extractor = cvlet.extractors(ie);
                cvlet.current_model = cvlet.models(im);

                warning('off', 'MATLAB:rmpath:DirNotFound');
                rmpath(genpath(cvlet.extractors_path));
                rmpath(genpath(cvlet.models_path));
                warning('on', 'MATLAB:rmpath:DirNotFound');

                addpath([cvlet.extractors_path, ...
                  cvlet.current_extractor.path]);
                addpath([cvlet.models_path, ...
                  cvlet.current_model.path]);

                cvlet.ident_or_verif = isequal( ...
                  cvlet.current_model.mode, 'verification');
                cvlet.bulk_postprocess = ...
                  cvlet.current_extractor.bulk_postprocess;

                cvlet_run(cvlet.current_database, ...
                  cvlet.current_extractor, cvlet.current_model);
```

```
                i = i + 1;
            end
        end
    end
end
```

Slika 3.3: Poenostavljena Matlab koda delovanja zunanje zanke izvajanja, ki skrbi za kombiniranje vseh modelov, metod za pridobivanje značilk in baz, ki jih je uporabnik nastavil v začetni konfiguraciji

Orodje za razpoznavo uhljev omogoča verifikacijo in identifikacijo. Pri identifikaciji med množico možnih pripadnosti razredu za vsak primer ugotavljamo, kateremu razredu pripada. Pri verifikaciji pa ugotavljamo ali posamezen primer pripada v izbran razred ali ne. Na področju razpoznave oseb na podlagi uhljev to pomeni, da pri identifikaciji za vsako sliko uhlja v naboru obravnavane množice ugotavljamo, za katero osebo gre, pri verifikaciji pa le preverjamo ali gre za osebo za katero se ta oseba izdaja ali ne.

Orodje za razpoznavo uhljev med delovanjem sproti shranjuje določene podatke, kar omogoča hitrejše delovanje ob naslednjih zagonih. Vsi vmesni in končni rezultati (vsi izhodi orodja za razpoznavo uhljev) se shranjujejo v mapo `_output`. Vsebina mape ne vpliva na končne rezultate in se jo lahko vedno izbriše, če uporabnik tako želi.

Mape znotraj mape `_output` za shranjevanje so sledeče:

- `calculated_features`: V to mapo se shranjujejo izračunane značilke uhljev. Shranijo se v formatu `CSV` (angl. comma separated values). Ime datoteke je oblike `baza-metoda`, kjer `baza` predstavlja ime mape baze, ki je bila uporabljena, `metoda` pa ime uporabljene metode za pridobivanje značilk uhljev.
  Razlog za takšno shranjevanje (in ne zgolj vmesno shranjevanje v Matlabovem formatu `.mat`, ki je hitrejše) je v zagotavljanju neodvisnosti. Tako shranjene značilke lahko raziskovalci uporabijo v drugih orodjih ali okoljih, ki lahko potencialno nudijo boljše odločitvene modele. Takšen primer bi bil že vnaprej pripravljen zmogljiv odločitveni model v na primer Pythonu ali katerem drugem jeziku oziroma okolju. Značilke shranjene v takšnem formatu lahko uporabimo tudi v povsem drugačnih sistemih, na primer IBM Watsonu [19] ipd. Vsekakor

se nam je zdelo pomembno ločiti ta del in omogočiti neodvisno shranjevanje značilk.

Zaradi časovne potratnosti takšnega shranjevanja se uporabnika ob pričetku zagona orodje za razpoznavo uhljev vpraša, če želi takšno shranjevanje (vidno na sliki 3.4). Branje že obstoječih shranjenih značilk se izvede brezpogojno – če želimo to preprečiti in prisiliti orodje za razpoznavo uhljev v ponovni izračun značilk, je potrebno izbrisati ali preimenovati pripadajočo `CSV` datoteko.

- `classifiers`: Mapa vsebuje naučene odločitvene modele v Matlabovem formatu `.mat`. Razloga za shranjevanje naučenih odločitvenih modelov sta dva: modela ni potrebno vedno na novo učiti in možnost deljenja že naučenega modela s skupnostjo.
- `temp`: Mapa vsebuje prebrane podatkovne baze v Matlabovem formatu za hrambo `.mat`. Edini razlog za takšno shranjevanje je v hitrosti izvajanja naslednjih poskusov. Nalaganje te datoteke iz diska je hitrejše kot branje vsake posamezne slikovne datoteke in gradnje seznama.
- `results`: V mapo rezultatov se shranijo tako skupni rezultati v tekstovni obliki kot tudi rezultati posameznih kombinacij podatkovnih baz, metod za pridobivanje značilk uhljev in odločitvenih modelov v formatu JSON. Vsa imena so sestavljena iz datuma zagona postopka evalvacije v formatu oblike `LLLL-MMDD-hhmm-ssfff` z vodilnimi ničlami[4]. S tem je istočasno zagotovljena unikatnost (da ne prihaja do neželenih prepisovanj obstoječih rezultatov) in preglednost. Skupno poročilo se shrani v korensko mapo rezultatov, rezultati posameznih kombinacij se shranjujejo v podmape z imeni po sledečem pravilu: `baza-metoda-klasifikator-način`, kjer `baza` predstavlja ime mape, v kateri se uporabljena baza nahaja, `metoda` ime mape, kjer se nahaja uporabljena metoda za pridobivanje značilk uhljev, `klasifikator` ime mape, kjer se nahaja uporabljen odločitveni model in `način`, ki lahko zavzema vrednosti `ide` (način identifikacije) ali `ver` (način verifikacije).

Na sliki 3.4 je prikazan primer zagona orodja za razpoznavo uhljev na podatkovni bazi uhljev IIT Delhi I z metodama Histogrami orientiranih gradientov (HOG) in Rotacijsko invariantni metodi kvantizacije lokalne faze z odločitvenim modelom k-najbližjih sosedov. Uporabniku se na vrhu izpiše identifikacijska šte-

[4]L – leto, M – mesec, D – dan, h – ura, m – minuta, s – sekunda, f – milisekunda

vilka zagona `Run ID`, ki ustreza imenom datotek z rezultati. Prikazan je zagon z dvema kombinacijama, vendar lahko poženemo poljubno število kombinacij primerjav.

Na sliki 3.5 je prikazana glavna drevesna struktura map orodja.

```
CVLET
+---core
+---databases
|   \---cvledb_1
+---extractors
|   +---cvlet-rilpq
|   +---cvlet_dsift
|   +---cvlet_fusion_h-n
    ...
|   \---SCAFFOLD
+---libraries
|   +---jsonlab
|   \---vlfeat-0.9.20
+---models
|   +---cvlet_knn
|   +---cvlet_random
|   +---cvlet_svm
|   \---SCAFFOLD
\---_output
    +---calculated_features
    +---models
    +---results
    \---temp
```

Slika 3.5: Drevesna struktura map orodja za razpoznavo uhljev

3.2.1 Priprava parametrov

Za začetek uporabe ni potrebno spreminjati konfiguracije oziroma spreminjati privzetih vrednosti. Spremembe pa so potrebne, če želimo uporabiti svoj odločitveni

Slika 3.4: Primer zagona orodja za razpoznavo uhljev

model, poljubno bazo uhljev, novo metodo za pridobivanje značilk uhljev. Za to moramo razumeti vrednosti v konfiguracijski datoteki in so opisane v nadaljevanju:

- `databases`
  Seznam struktur (Matlab `struct`), ki vsebujejo informacije o podatkovnih bazah uhljev, ki jih želimo uporabiti. Vsaka ima dva parametra:
  - `path`: Ime mape v kateri se naha podatkovna baza uhljev.
  - `name`: Ime podatkovne baze uhljev. Ime se prikaže v končnih rezultatih in poročilu.
- `extractors`
  Seznam struktur (Matlab `struct`), ki vsebujejo informacije o metodah za pridobivanje značilk uhljev v mapi, kjer ima vsaka tri parametre:
  - `path`: Ime mape v kateri se nahajajo datoteke funkcij za pridobivanje značilk uhljev.
  - `name`: Ime metode za pridobivanje značilk uhljev. Ime se prikaže v končnih rezultatih in poročilu.
  - `bulk_postprocess`: Logična vrednost, ki določa obnašanje poprocesiranja. Če je vrednost 1 (true), se bo pripadajoča funkcija za poprocesiranje klicala za vse vektorje značilk uhljev skupaj, sicer za vsakega posebej. Tako lahko v primeru, ko so vsi vektorji enako veliki (torej imamo matriko značilk), opravljamo transformacije na celotni matriki, kar je v Matlabu hitreje, kot če enak proces izvajamo iterativno za vsako sliko posebej.
- `models`
  Seznam struktur (Matlab `struct`), ki vsebujejo informacije o uporabljenih odločitvenih modelih, vsaka vsebuje tri parametre:
  - `path`: Ime mape, v kateri se nahajajo datoteke funkcij za odločitveni model.
  - `name`: Ime odločitvenega modela za klasificiranje oseb na podlagi pridobljenih značilk uhljev.
  - `mode`: Niz, ki definira način evalvacije: verifikacija ali identifikacija. Niz lahko vsebuje vrednost `identification` ali `verification`.
- `log_level`
  Številska vrednost, ki lahko zavzema tri vrednosti: 0, 1 ali 2 in določa nivo beleženja delovanja v konzoli. Nivo 0 ne prikazuje ničesar, pri nivoju 1 se prikazujejo standardni opisi, nivo 2 nudi podrobno poročanje in je namenjen

morebitnemu razhroščevanju. Nastavljanje tega parametra ne vpliva na izpis končnih rezultatov.

- `crossvalid`
  Struktura (Matlab `struct`), ki določa način izračuna rezultatov:
  - `method`: Možni sta dve vrednosti: `Kfold` (k-kratno prečno preverjanje) ali `HoldOut` (pridržanje dela množice). Pri k-kratnem prečnem preverjanju podatke razdelimo na k delov. Eden izmed delov predstavlja testno množico, ostali sestavljajo učno. Postopek se ponovi k-krat, rezultati se povprečijo. Pri pridržanju dela množice, določen (naključni) del vzamemo za testni del, ostalo pa za učni. Če želimo dobiti dovolj dobre ocene natančnosti moramo postopek ponoviti in rezultate ustrezno povprečiti.
  - `factor`: Parameter določa razmerje med učno in testno množico. Če je vrednost `method` nastavljena na `Kfold`, se pričakuje celo število – pri vrednosti 3 to pomeni, da se podatke razdeli v tri skupine in nato pri trikratni ponovitvi ena predstavlja testno množico, drugi dve sestavljata učno. Če je vrednost `method` nastavljena na `HoldOut` pa se pričakuje decimalno število, ki predstavlja direktno razmerje testne množice – vrednost $0,3$ pomeni, da bo 30% vseh primerov uporabljenih za testno množico, 70% pa za učno.
- `cache`
  Logična vrednost (0 ali 1), ki definira ali orodje sme delati začasno hranjenje podatkov. Med te podatke spada prebrana podatkovna baza, izračunane vrednosti vektorjev značilk uhljev in naučen klasifikator.

3.2.2 Priprava podatkovne baze uhljev

Za uporabo lastne podatkovne baze je dovolj, da vse slike skopiramo v poljubno poimenovano mapo znotraj mape `databases`, ki se nahaja znotraj korenske mape orodja za razpoznavo uhljev. Edina zahteva je, da so slike uhljev razvrščene po podmapah za vsako osebo posebej.

Podprti so sledeči formati slik: PNG, JPG in BMP.

Branje podatkov o anotacijah se izvaja preko JSON datotek, ki lahko vsebujejo poljubne parametre, vendar morajo sestavljati veljaven format zapisa JSON. Vsaka JSON datoteka mora biti poimenovana `annotations.json` in mora biti shranjena znotraj vsake podmape osebe.

Za lažjo predstavo si lahko končni uporabnik pogleda strukturo priložene podatkovne baze uhljev.

Orodje za razpoznavo uhljev med delovanjem vsebino podatkovne baze shrani v pomnilnik in na datotečni sistem v obliki Matlab `.mat` datoteke, saj to pospeši izvajanje ob naslednjem zagonu orodja za razpoznavo uhljev. Če uporabnik tega ne želi, lahko to izklopi s pomočjo parametra `cache` v osnovni konfiguracijski datoteki.

3.2.3 Priprava metode za pridobivanje značilk uhljev

Za lažje dodajanje novih metod za pridobivanje značilk uhljev je na voljo mapa `SCAFFOLD`, ki vsebuje demonstracijske datoteke s funkcijami, ki so potrebne za delovanje metod za pridobivanje značilk uhljev. Funkcije vsebujejo zaglavja s potrebnimi funkcijskimi parametri in pripadajočimi komentarji. Od uporabnika se pričakuje le, da vstavi lastno programsko kodo za izvajanje želenih funkcij.

Funkcije so sledeče:

- `cvlet_init_extractor`: Funkcija ni obvezna. Vsebuje poljubno programsko kodo, ki se izvede le enkrat in sicer pred postopkom pridobivanja značilk uhljev. Mišljena je za primere, ko pred izvajanjem potrebujemo naložitev določene knjižnice ali vključitev zunanje kode, nastavitev globalnih vrednosti ali kaj podobnega in tudi nima predvidenih vhodnih in izhodnih parametrov. Podpis funkcije je prikazan na sliki 3.6.
- `cvlet_preprocess`: Funkcija ni obvezna. Namen je predpriprava slike uhlja in se izvede za vsako sliko uhlja posebej pred samim postopkom pridobivanja značilk. Primer je pretvorba barvne slike v sivinsko. Podpis funkcije je prikazan na sliki 3.7.
- `cvlet_features_extract`: Funkcija je obvezna. Če funkcija ni prisotna, orodje za razpoznavo uhljev sproži napako in ustavi izvajanje. Gre za glavna funkcijo, ki je namenjena pridobivanju značilk. Na vhod dobi sliko uhlja, na izhod pa da vektor značilk. Anotacijski podatki na vhodu v funkcijo služijo za potencialno boljše pridobivanje značilk uhlja (na primer mesto tragusa in smer uhlja). Podpis funkcije je prikazan na sliki 3.8.
- `cvlet_postprocess`: Funkcija ni obvezna, služi pa obdelavi vektorjev značilk uhljev po izvedenem procesu pridobivanja značilk. Funkcija se lahko izvaja v dveh načinih, glede na parameter `bulk_postprocess`, nastavljen v osnovni kon-

figuracijski datoteki. Če je nastavljen na 0 bo sprožil funkcijo po vsaki pridobitvi značilk uhljev, če pa na 1 pa šele na koncu obdelave vseh slik uhljev. V prvem primeru je vhod v funkcijo vektor značilk enega uhlja, v drugem primeru pa matrika značilk vseh uhljev. Analogno velja za izhod funkcije, ki tako predstavlja ali obdelan vektor značilk enega uhlja ali obdelano matriko značilk vseh uhljev. Podpis funkcije je prikazan na sliki 3.9.

Opisane funkcije niso nujno edine, so le tiste, ki jih orodje za razpoznavo uhljev pokliče, če so prisotne. Uporabnik lahko uporabi poljubno število pomožnih datotek oziroma funkcij. Na prikazanih slikah 3.6, 3.7, 3.8 in 3.9 so komentarji slovenjeni – v orodju za razpoznavo uhljev so vsi komentarji v angleščini, zaradi namena rabe na raziskovalnem področju, kjer prevladuje angleščina.

```
function cvlet_init_extractor()
    % funkcija namenjena ekratnemu izvajanju pred
    % postopkom pridobivanja znacilk

    % KODA UPORABNIKA
end
```

Slika 3.6: Podpis neobvezne funkcije `cvlet_init_extractor`

```
function [I, I_annotations] = cvlet_preprocess(I, I_annotations)
    % postopek preprocesiranja se izvede za
    % vsako sliko uhlja I posebej
    %
    % Vhod:
    %    I              = slika uhlja
    %    I_annotations  = anotacijski podatki za trenutno
    %                     sliko uhlja I
    %
    % Izhod:
```

```
    %    I              = nova slika uhlja
    %    I_annotations  = novi anotacijski podatki
    %                     za trenutno sliko uhlja I

    % KODA UPORABNIKA
end
```

Slika 3.7: Podpis neobvezne funkcije `cvlet_preprocess`

```
function [features] = cvlet_features_extract(I, I_annotations)
    % pridobimo znacilke za trenutno sliko uhlja I
    %
    % Vhod:
    %    I             = predprocesirana slika uhlja
    %    I_annotations = anotacijski podatki za trenutno
    %                    sliko uhlja I
    %
    % Izhod:
    %    features      = vektor znacilk pridobljenih iz
    %                    slike uhlja I

    % KODA UPORABNIKA
 end
```

Slika 3.8: Podpis obvezne funkcije `cvlet_features_extract`

```
function [features] = cvlet_postprocess(features)
    % postprocesiranje za posamezen vektor
    % znacilk ali za vse skupaj - odvisno od
    % parametra bulk_postprocess v konfiguraciji
    %
    % Vhod:
```

```
    %    features = seznam znacilk za vse uhlje ALI
    %               znacilke le za trenuten uhelj
    %
    % Izhod:
    %    features = nove znacilke
    %
    % Ce je vkjucen parameter bulk_postprocess,
    % je to primer zanke, ki jo lahko uporabite:
    % for i = 1:size(features,1)
    %   [features(i, :).feature] = ...
    %               smth(features(i, :).feature);
    % end

    % KODA UPORABNIKA
end
```

Slika 3.9: Podpis neobvezne funkcije `cvlet_postprocess`

3.2.4 Priprava odločitvenega modela

Za lažje dodajanje novih metod za odločanje je, tako kot pri metodah za pridobivanje značilk uhljev, na voljo mapa `SCAFFOLD`, ki vsebuje demonstracijske datoteke s funkcijami, ki so potrebne za delovanje odločitvenih modelov. Funkcije vsebujejo zaglavja s potrebnimi funkcijskimi parametri in pripadajočimi komentarji. Od uporabnika se tudi tukaj pričakuje le, da vstavi lastno programsko kodo za izvajanje želenih funkcij.

Funkciji sta sledeči:

- `cvlet_init_model`: Funkcija ni obvezna. Vsebuje programsko kodo, ki se izvede le enkrat in sicer pred postopkom učenja in klasificiranja podatkov. Tako kot pri metodah za pridobivanje značilk uhljev je mišljena za primere, ko pred izvajanjem potrebujemo enkratno izvajanje neke programske kode in tudi nima predvidenih vhodih in izhodnih parametrov. Funkcija je ločena od `cvlet_evaluate` zaradi večje preglednosti in ločenosti inicializacijskega dela od samega delovanja jedra. Podpis funkcije je prikazan na sliki 3.10.
- `cvlet_evaluate`: Funkcija je obvezna. Če funkcija ni prisotna, orodje za raz-

poznavo uhljev sproži napako in ustavi izvajanje. Gre za glavna funkcijo, ki je namenjena klasificiranju testnih primerov v razrede. Funkcija dobi na vhod matriko učnih in testnih podatkov ter seznam pripadnosti učnih podatkov posameznim razredom. Podpis funkcije je prikazan na sliki 3.11.

```
function cvlet_init_model()
    % funkcija namenjena ekratnemu izvajanju pred
    % postopkom klasifikacije

    % KODA UPORABNIKA
end
```

Slika 3.10: Podpis neobvezne funkcije `cvlet_init_model`

```
function results = cvlet_evaluate(X_train, y_train, X_test)
    % predvidevanje rezultatov na podlagi X_train in y_train
    %
    % Vhod:
    %    X_train = matrika ucnih podatkov
    %    y_train = seznam pripadnosti ucnih podatkov
    %              posameznim razredom
    %    X_test  = matrika testnih podatkov
    %
    % Izhod:
    %    results = seznam napovedanih vrednosti (y_test)

    % KODA UPORABNIKA
end
```

Slika 3.11: Podpis obvezne funkcije `cvlet_evaluate`

3.2.5 Izpis in shranjevanje rezultatov

Izpis rezultatov je zadnji korak orodja za razpoznavo uhljev. Del rezultatov (splošna natančnost) se izpiše že v konzolo, podrobnejši rezultati pa v datoteke v formatu JSON. Podrobnejši rezultati se zapišejo v pripadajoče mape, ki so sestavljene iz imena baze, metode in odločitvenega modela, po načelu, ki je opisano v poglavju 3.2. Imenu mape je dodana še tričrkovna oznaka načina testiranja, to je `ide` za način identifikacije in `ver` za način verifikacije. Ime same datoteke je sestavljeno iz omenjenega `RunID` parametra.

Mer za ocenjevanje zmogljivosti metod za razpoznavo oseb na podlagi biometričnih modalnosti je veliko [20]. V orodje smo vključili naslednje:

- natančnost
- števila: pravilno pozitivnih, pravilno negativnih, napačno pozitivnih, napačno negativnih, vseh primerov, vseh negativnih primerov, vseh pozitivnih primerov
- specifičnost (angl. specificity)
- senzitivnost (angl. sensitivity)
- razmerji napačno negativnih in napačno pozitivnih.

Poglavje 4

Pridobivanje značilk uhljev

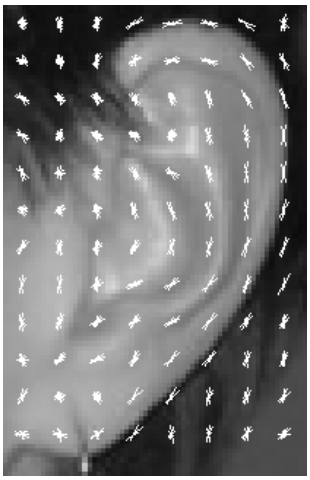
V nadaljevanju je podanih 8 metod za pridobivanje značilk uhljev, seveda pa to niso vse. Naštete so le tiste, ki so bile vključene v testiranje in so vključene v razvito orodje za razpoznavo uhljev.

Metode so bile izbrane zarad obetavnosti glede na rabo pri drugih biometričnih modalnostih v literaturi (kvantizacija lokalne faze, rotacijsko invariantna metoda kvantizacije lokalne faze, fuzija), zaradi preprostosti in dostopnosti (normirane slike, histogrami orientiranih gradientov), ali pogoste rabe v literaturi (skalirno invariantna transformacija značilk, gosta skalirno invariantna transformacija značilk, maksimalno stabilni ekstremi regij, fuzija).

4.1 Direktna raba normiranih slik

Pri načinu direktne rabe slik se velikost vsake slika normira na velikost 32×32 slikovnih točk in pretvori v sivinsko sliko. Tako predpostavljamo, da je odločitveni model dovolj zmogljiv, da dovolj dobro klasificira obravnavane primere.

Ta način je posebej primeren v kombinaciji z globokimi nevronskimi mrežami, na primer s konvolucijskimi nevronskimi mrežami (angl. convolutional neural network – cNN), saj so take metode zaradi velike globine same sposobne najti ustrezne vzorce [21]. Direktna raba normiranih slik je sicer delovala solidno tudi v primerih k-najbližjih sosedov in z metodo podpornih vektorjev, kot je vidno v poglavju 6.



Slika 4.1: Prikaz opisnikov metode histogramov orientiranih gradientov

4.2 Histogrami orientiranih gradientov

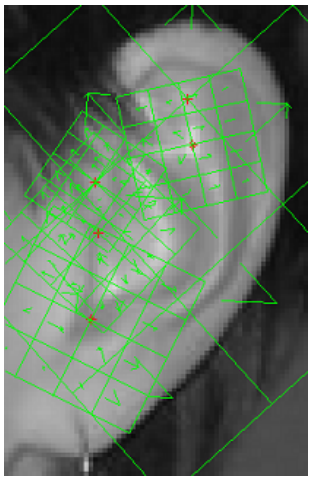
Metoda histogramov orientiranih gradientov (angl. histogram of oriented gradients – HOG) [22] je uporabna, ko uporabljamo slike z različnimi osvetlitvami [22, 23].

Prvi korak pri procesu pridobivanja značilk je pretvorba v sivinsko sliko fiksnih dimenzij (v našem primeru je to 100×100). Slika je nato razdeljena v enakomerna področja, nad katerimi se nato izračunajo HOG opisniki. Primer takšnega izračuna opisnikov je prikazan na sliki 4.1.

Zaradi enakomerne velikosti začetnih področij za izračun opisnikov dobimo pri fiksni velikosti slike vedno enako velike vektorje. To pomeni, da lahko takšne opisnike direktno sestavimo v matriko vhodnih podatkov v odločitveni model, brez potrebe po rabi metod za krajšanje vektorjev.

Prednost tega, da ni potrebe po krajšanju vektorjev je v tem, da lahko pri krajšanju vektorjev vpeljemo napake – izbrišemo pomembne vrednosti, vrednosti režemo pristransko ipd. To lahko pomeni, da je končna zmogljivost manjša, kot če začetne slike razširimo ali skrčimo na fiksno velikost.

Če smo vektorje prisiljeni rezati (kot je to primer pri metodi SIFT v nadaljevanju) lahko uporabimo več metod. Očitno je, da enostavno rezanje vektorja značilk na enem koncu ni najboljša izbira, bolje je, če uporabimo kaj naprednejšega, kot je na primer tako imenovana metoda voditeljev (angl. k-means clustering), ki smo jo uporabili tudi mi.



Slika 4.2: Skalirno invariantna tranformacija značilk: prikaz izračunanih orientacij na regijah velikost 4×4

4.3 Skalirno invariantna tranformacija značilk

Metoda skalirno invariantne transformacije značilk (angl. scale-invariant feature transform – SIFT) [24] je zanimiva zaradi robustnosti na prostorske transformacije [25, 26]. Bila je že uporabljena tudi na področju razpoznave uhljev [26–28].

Tudi tu v prvem koraku sliko pretvorimo v sivinsko (korak 1). Sledijo koraki detekcije lokalnih ekstremov (korak 2), lokalizacije ključnih točk (korak 3), dodelitve orientacije (korak 4), pridobitev opisnikov ključnih točk (korak 5). Dodali smo še šesti korak: metodo voditeljev (angl. k-means clustering) [29,30] za zmanjšanje dolžine vektorjev značilk (pridobljenih opisnikov).

Pri korakih 2, 3, 4 in 5 smo si pomagali s knjižnico VLFeat [31].

Postopek korakov 3 in 4 je prikazan na sliki 4.2, kjer so z zeleno označen izračunane orientacije na regijah velikosti 4×4.

Glavna težava metode skalirno invariantne transformacije značilk (in vseh metod, ki se opirajo na ključne točke pri pridobivanju značilk) pri rabi za razpoznavo oseb na podlagi biometričnih podatkov uhljev je v izbiri teh točk. Pri primerjavi razredov slik (na primer razločevanje slik, ki vsebujejo nek predmet, od slik, ki jih ne) to zadošča, pri primerjavi uhljev oseb pa se pričakuje slabše delovanje od implementirane metode RI-LPQ.

4.4 Gosta skalirno invariantna tranformacija značilk

Metoda goste skalirno invariantne tranformacije značilk (angl. dense scale-invariant feature transform) je izpeljanka že opisane metode skalirno invariantne tranformacije značilk.

Pri metodi skalirno invariantne tranformacije značilk se število in lokacije ključnih točk (korak 2) spreminjajo glede na osvetlitev oz. že manjše spremembe v slikah. Pri metodi goste skalirno invariantne tranformacije značilk to lahko kontroliramo [32, 33].

4.5 Maksimalno stabilni ekstremi regij

Metoda maksimalno stabilnih ekstremov regij (angl. maximally stable extremal regions – MSER) [34] deluje na principu binarizacije slike. Prednost metode je njena računska preprostost in odpornost na enakomerne spremembe osvetlitve [34].

Metoda deluje na principu izbire regij, ki so najbolj stabilne – če bi pogledali sliko bi bila to območja, ki so najbolj gladka. Take regije se izbere tako, da se sliko upragovi, nato pa se pri iterativnem zviševanju praga ugotavlja, katera območja se ne spreminjajo oziroma se spreminjajo malo.

4.6 Kvantizacije lokalne faze

Metoda kvantizacije lokalne faze (angl. local phase quantization – LPQ) temelji na diskretni Fourierjevi tranformaciji (DFT), ki se računa za vsak kvadratni del slike posebej [35]. Zmogljivost LPQ metode izhaja iz dejstva, da so vektorji, ki jih pridobimo z računanjem LPQja robustni glede na zameglitev [36]. To je pri uhljih še posebej obetavno, saj imamo velikokrat opravka z dokaj majhnimi slikami – popačenje, ki nastane pri rabi majhne slike, pa je analogno zameglitvi. To je hkrati tudi glavna lastnost metode LPQ v primerjavi z metodo lokalnih binarnih vzorcev (angl. local binary pattern – LBP) [37].

Po izračunu DFTja, se seštejejo vse izračunane vrednosti za vsak kvadrat posebej, ki so večje ali enake 0. Vsote se nato sestavijo v histogram, ki šele nato služi

kot vektor značilk posameznega uhlja.

4.7 Rotacijsko invariantna metoda kvantizacije lokalne faze

Rotacijsko invariantna metoda kvantizacije lokalne faze (angl. rotation invariant local phase quantization – RILPQ) [38] je nadgrajena metoda kvantizacije lokalne faze.

V osnovi deluje enako kot metoda kvantizacije lokalne faze, le da pred pričetkom izvajanja naredimo oceno lokalne karakteristične orientacije (angl. local characteristic orientation), ki nato služi kot osnova za rotacijo okolice pred izračunom kvantizacije lokalne faze.

4.8 Fuzija

Motivacija za kombiniranje vektorjev značilk leži v dejstvu, da določene metode za pridobivanje značilk slabo izrazijo nekatere atribute. S kombiniranjem večih različnih značilk uhljev zajamemo večji nabor opisov uhljev, kot bi jih sicer. Naloga odločitvenega modela pa je potem, da ustrezno kaznuje ali nagradi določene atribute.

Fuzijo lahko izvajamo na večih nivojih [39]:

- na nivoju zaznave
- na nivoju pridobivanja značilk
- na nivoju ugotavljanja pripadnosti
- na odločitvenem nivoju.

Implementirali smo fuzijo na nivoju značilk, zaradi obetavnosti in dobrih možnosti kombiniranja [39], in sicer za naslednje kombinacije:

- rotacijsko invariantna kvantizacija lokalne faze in histogram orientiranih gradientov
- rotacijsko invariantna kvantizacija lokalne faze in normirane slike
- histogram orientiranih gradientov in normirane slike

- rotacijsko invariantna kvantizacija lokalne faze, histogram orientiranih gradientov in normirane slike.

Naštete kombinacije smo izbrali po testiranju vseh do sedaj obravnavanih metod za pridobivanje značilk in izbrali tiste tri, ki so dosegale najboljše rezultate. Fuzijske značilke so bile zgrajene tako, da so se vektorsko združile značilke posameznih metod, nato pa so te združene značilke služile kot vhod v odločitvene modele.

Poglavje 5

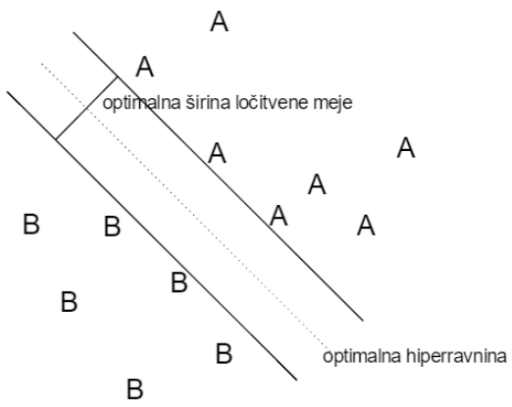
Uporabljeni odločitveni modeli

Pri razpoznavi oseb na podlagi biometričnih podatkov uhljev smo uporabili dva odločitvena modela: metodo podpornih vektorjev in metodo k-najbližjih sosedov. Uporabili smo Matlab implementacijo s privzetimi vrednostmi. Pri odločitvenih modelih je še veliko prostora za izboljšave, saj glavni fokus pričujoče naloge v drugem delu ni bil na odločitvenih modelih, temveč na metodah za pridobivanje značilk.

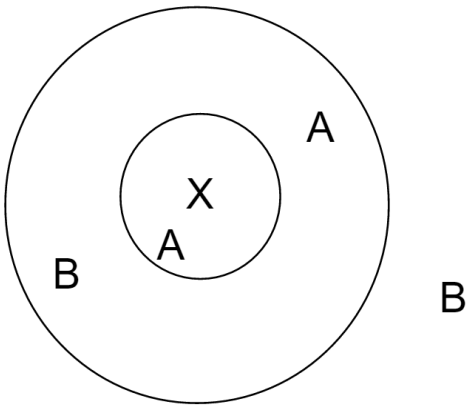
5.1 Metoda podpornih vektorjev

Metoda podpornih vektorjev (angl. support vektor machine – SVM) [40] temelji na določitvi razmejitvene hiperravnine tako, da je oddaljenost primerov od te ločitvene meje čim večja, kot to prikazuje slika 5.1. Čeprav gre v osnovi za linearni klasifikator, lahko klasificira tudi nelinearne primere tako, da vhodne vektorje obravnava večdimenzionalno. Ta obravnava se ne zgodi z direktno transformacijo vseh vektorjev v večdimenzionalne, saj je to računsko zahtevno, temveč z ustrezno rabo jedrne funkcije [40].

Glavna težava metode podpornih vektorjev, na katero smo naleteli tudi med nastajanjem tega dela, je v računski zahtevnosti, sploh v primerjavi z metodo k-najbližjih sosedov. Metodo podpornih vektorjev smo izbrali zaradi zmogljivosti in široke rabe v literaturi, vendar se je v končnih testih, v splošnem, odrezala slabše od metode k-najbližjih sosedov. Metoda podpornih vektorjev ima težave s šumnimi podatki in podatki, ki niso homogeni. To je gotovo eden izmed vzrokov,



Slika 5.1: Prikaz SVM klasifikacije in optimalne ločitvene meje



Slika 5.2: Prikaz kNN klasifikacije v primerih $k = 1$ in $k = 3$

da se SVMji tako pri naših testih, kot v literaturi pri konkretni nalogi klasifikacije na podlagi uhljev niso izkazali. Če bi uhlje predhodno detektirali in jih ločili od preostanka slike (imeli bi manj šumnih podatkov), je za pričakovati, da bi se natančnost klasifikacije metode podpornih vektorjev izboljšala glede na druge.

Metoda podpornih vektorjev v osnovi klasificira binarno – torej le v dva razreda. Za uporabo večrazredne klasifikacije v primeri identifikacije moramo opraviti primerjavo vseh možnih kombinacij.

5.2 Metoda k-najbližjih sosedov

Metodo k-najbližjih sosedov (angl. k-nearest neighbors – kNN) lahko uporabljamo tako za klasifikacijo kot regresijo. Glavna prednost te metode je preprostost in računska nezahtevnost in je bila zato tudi izbrana, saj je zahtevnost učnega procesa nična. Glavna težava pa je občutljivost na visoko dimenzionalnost – to je veliko število atributov glede na število primerov pri vhodnih podatkih.

Osnovna ideja temelji na klasificiranju na podlagi bližine najbližjih sosedov. Parameter k določa število elementov oz. sosedov znotraj območja, ki določa razred trenutnega primera. V testih smo uporabili fiksno vrednost $k = 1$ (vmesni testi so pokazali boljše rezultate kot z večjimi vrednostmi k), vendar bi tudi izbor parametra k lahko vpeljali v postopek učenja.

Na sliki 5.2 sta prikazana primera klasifikacije, ko je $k = 1$ (v razred zajamemo

notranji A) in ko je $k = 3$ (v razred zajamemo tudi drugi A, vendar hkrati tudi neželeni B).

Poglavje 6

Rezultati

Pri testiranju je bilo uporabljeno 10-kratno prečno preverjanje. To pomeni, da so bile množice podatkov razdeljene v desetine, kjer je pri vsaki iteraciji $\frac{1}{10}$ podatkov predstavljalo testno množico, ostalih $\frac{9}{10}$ pa učno. Izvajanje je bilo ponovljeno $10\times$, končni rezultati pa so povprečeni. Razlog za k-kratno prečno preverjanje je ta, da zagotovimo, da se bodo vedno uporabili vsi primeri iz množice slik. Razlog za izbiro ravno faktorja 10 pa je v pogostosti uporabe v sorodnih delih in literaturi ter dobrim razmerjem med varianco in kompleksnostjo izvajanja [41]. Bolj ko faktor k povečujemo, bolj zmanjšujemo pričakovano varianco rezultatov, vendar hkrati tudi podaljšujemo čas izvajanja. Pri določeni vrednosti se varianca ne zmanjšuje več opazno, čas izvajanja pa se vseeno povečuje.

Preverjanja so bila zagnana tako v načinu identifikacije kot v načinu verifikacije. Med levimi in desnimi uhlji nismo razlikovali, čeprav bi, pri uporabi baze WPUTEDB, to izboljšavo lahko vpeljali. Čeprav ima večina ljudi uhlje simetrične [1], je število asimetričnih primerov gotovo dovolj veliko, da bi bilo možno opaziti razliko v delovanju.

V tabelah 6.1, 6.2 in 6.3 so okrajšave sledeče:

- BSIF: binarizirane statistične značilke slik
- CVLEDB: naša podatkovna baza uhljev
- Delhi I R: IIT Delhi baza I z neprocesiranimi slikami
- Delhi I P, Delhi II P: IIT Delhi bazi I in II s procesiranimi silkami
- DSIFT: gosta skalirno invariantna tranformacija značilk
- Fu H-N: fuzija histogramov orientiranih gradientov in normiranih slik uhljev

- Fu R-H: fuzija rotacijsko invariantne kvantizacije lokalne faze in histogramov orientiranih gradientov
- Fu R-H-N: fuzija rotacijsko invariantne kvantizacije lokalne faze, histogramov orientiranih gradientov in normiranih slik uhljev
- Fu R-N: fuzija rotacijsko invariantne kvantizacije lokalne faze in normiranih slik uhljev
- HD: Hammingova razdalja
- HOG: histogram orientiranih gradientov
- ICP: klasifikator skalarnega produkta
- kNN: metoda k-najbližjih sosedov
- LPIC: metoda lokalnih glavnih neodvisnih komponent
- LPQ: kvantizacije lokalne faze
- MSER: maksimalno stabilni ekstremi regij
- NORM: normirane slike uhljev
- PCA: metoda glavnih komponent
- RILPQ: rotacijsko invariantna kvantizacije lokalne faze
- SIFT: skalirno invariantna tranformacija značilk
- SVM: metoda podpornih vektorjev
- WPUTEDB: baza slik uhljev WPUTEDB.

Kot je razvidno iz tabele 6.1 so se pri identifikaciji najbolje odrezale sledeče metode: histogram orientiranih gradientov, rotacijsko invariantna kvantizacije lokalne faze, normirane slike in fuzijski metodi R-H-N (rotacijsko invariantna kvantizacije lokalne faze, histogram orientiranih gradientov in normirane slike) in R-N (rotacijsko invariantna kvantizacije lokalne faze in normirane slike).

Pri večini je bil klasifikator k-najbližjih sosedov boljši od metod podpornih vektorjev. Razlog je verjetno v tem, da odločitvenih modelov nismo optimizirali in je tu gotovo še veliko možnosti za izboljšave. Povsod kjer ni navedeno eksplicitno, se za natančnost posamezne metode šteje tisti klasifikator, ki je boljši.

Pregled natančnosti po podatkovnih bazah je sledeč:

- CVLEDB: Najbolje se je odrezal histogram orientiranih gradientov in sicer s $76,74\%$ natančnostjo. Rotacijsko invariantna kvantizacija lokalne faze se je odrezala slabo s $24,19\%$ natančnostjo, kar kaže na to, da se metoda ne obnaša dobro, ko so slike v splošnem zelo različne.

- IIT Delhi I R: Tu se je najbolje odrezala fuzija R-H-N, sledita ji rotacijsko invariantne kvantizacije lokalne faze z $79,01\%$ natančnostjo in histogram orientiranih gradientov z natančnostjo $72,21\%$
- IIT Delhi I P: Najbolje se je odrezala fuzija R-N z $92,7\%$ natančnostjo. Sledita ji normirane slike $(92,19\%)$ in fuzija R-H-N. Razlog za tako dobro natančnost normiranih slik gre verjetno iskati v tem, da je baza poprocesirana - slike so enako velike, rotacija uhljev je normalizirana, uhlji so centrirani. Pri naši podatkovni bazi, ki velja za najtežjo od obravnavanih, se ta metoda ni obnesla.
- IIT Delhi II P: Najboljša metoda je histogram orientiranih gradientov z $90,86\%$ natančnostjo. Sledi jih fuzija R-N z natančnostjo $87,64\%$.
- WPUTEDB: Tako kot pri IIT Delhi II P, je najboljša metoda histogram orientiranih gradientov z $81,96\%$ natančnostjo. Sledi jih fuzija R-N z natančnostjo $80,48\%$.

Razlog za splošno slab rezultat metod: skalirno invariantna tranformacija značilk in goste skalirno invariantna tranformacija značilk je, verjetno, v krajšanju vektorjev značilk s tako imenovano metodo voditeljev (k-means clustering), ki vnesejo v končne značilke napake.

Razlog za slab rezultat metode podpornih vektorjev pri metodah kvantizacije lokalne faze in metode maksimalno stabilnih ekstremov regij leži v lastnostih njunih značilk. Razlike vrednosti v opisnih vektorjih pridobljenih z metodo maksimalno stabilnih ekstremov regij so velikostno gledano zelo velike, tudi za faktor reda 10^6. Pri metodi kvantizacije lokalne faze so vse vrednosti opisnikov zelo majhne, veliko atributov je 0. Obe omenjeni lastnosti metode k-najbližjih sosedov ne zmotita, medtem ko metoda podpornih vektorjev na takšnih podatkih ne zna klasificirati. Tega bi se lahko znebili, če bi v postopku poprocesiranja vrednosti opisnikov zbližali oz. vrednosti ustrezno povečali.

Tabela 6.2 prikazuje natačnost klasifikacije v načinu verifikacije. Rezultati v načinu verifikacije so na videz boljši, vendar je to zgolj zato, ker imamo pri takem načinu veliko negativnih primerov in malo pozitivnih. Ne glede na to, kako slaba ali dobra je metoda za pridobivanje značilk, se bo odločitveni model naučil, da je večina primerov negativnih in bo tako tudi klasificiral. Komentarji na osnovi natančnosti v tabeli 6.1 tako veljajo tudi za tabelo 6.2, le da so razlike v načinu verifikacije slabše vidne zaradi splošno višjih vrednosti natančnosti.

Tabela 6.1: Natančnost klasifikacije pri identifikaciji [%]

		CVLEDB	Delhi I R	Delhi I P	Delhi II P	WPUTEDB
RILPQ	kNN	24,19	**79,01**	79,82	72,95	77,03
	SVM	26,37	42,39	50,91	35,1	58,33
LPQ	kNN	23,2	62,68	57,3	53,78	76,6
	SVM	8,71	1,27	0,8	0	3,45
HOG	kNN	**76,74**	72,21	91,18	**90,86**	**81,96**
	SVM	71,77	51,4	69,57	70,83	55,34
MSER	kNN	24,32	37,83	33,67	30,08	74,19
	SVM	18,78	1,42	1,22	0,38	2,55
SIFT	kNN	18,35	40,77	38,95	28,94	64,52
	SVM	17,41	26,17	22,11	17,78	49,5
Fu H-N	kNN	39,68	69,98	90,67	85,5	76,76
	SVM	40,01	30,11	78,61	39,2	66,4
Fu R-H	kNN	24,13	77,89	79,92	73,77	65,22
	SVM	22,55	39,76	60,66	35,22	31,7
Fu R-H-N	kNN	40,55	**79,51**	**91,48**	86,51	79,12
	SVM	25,22	40,83	85,12	42,55	63,83
Fu R-N	kNN	40,3	75,66	**92,7**	**87,64**	**80,48**
	SVM	35,96	38,38	79,7	40,1	60,66
DSIFT	kNN	24,88	55,68	56,29	51,26	64,8
	SVM	33,59	32,86	37,32	31,02	42,33
NORM	kNN	39,99	66,53	**92,19**	86,19	76,31
	SVM	55,34	50,5	73,23	58,89	59,21

Tabela 6.2: Natančnost klasifikacije pri verifikaciji [%]

		CVLDB	Delhi I R	Delhi I P	Delhi II P	WPUTEDB
RILPQ	kNN	99,35	100	99,99	99,99	99,99
	SVM	99,35	100	99,99	99,99	99,99
LPQ	kNN	99,32	100	99,98	99,99	99,99
	SVM	99,63	99,99	99,99	99,99	99,99
HOG	kNN	99,78	100	99,99	99,99	99,99
	SVM	99,79	100	100	99,99	99,99
MSER	kNN	99,46	99,99	99,98	99,99	99,99
	SVM	99,63	99,98	99,99	99,99	99,98
SIFT	kNN	99,42	99,99	99,97	99,99	99,99
	SVM	99,02	99,99	99,99	99,99	99,99
FU H-N	kNN	99,55	100	99,99	99,99	99,99
	SVM	99,6	99,99	100	99,99	99,98
FU R-H	kNN	99,46	100	99,99	99,99	99,99
	SVM	99,26	100	99,99	99,99	99,99
FU R-H-N	kNN	99,53	100	99,99	99,99	99,99
	SVM	99,69	100	100	100	99,99
FU R-N	kNN	99,53	100	99,99	99,99	99,99
	SVM	99,61	100	100	99,99	99,99
DSIFT	kNN	99,37	100	99,99	99,99	99,99
	SVM	99,43	99,99	99,99	99,99	99,99
NORM	kNN	99,58	100	99,99	99,99	99,99
	SVM	99,65	99,99	100	99,99	99,99

Tabela 6.3: Primerjava natančnosti najboljših metod za razpoznavo oseb na podlagi uhljev pri identifikaciji na bazah IIT Delhi I P in IIT Delhi II P [%]

metoda	klasifikator	Delhi I P	Delhi II P
LPIC [42]	ICP	97,6	97,2
BSIF descriptor [43, 44]	kNN	97,26	97,34
Non linear curvelet features [45]	kNN	97,77	96,22
2D quadrature filter [46]	HD	96,53	96,08
Log-Gabor [3]	kNN	96,27	95,93
1D quadrature filter [46]	HD	95,73	94,72
Fuzija R-N	kNN	**92,7**	**87,64**
Weighted Gabor orientation [3]	kNN	90,4	88,39
Gabor [3]	kNN	88,53	86,73
Gabor phase [3]	kNN	83,47	84,46
PCA [42]	ICP	82,6	79
RILPQ	kNN	**79,82**	**72,95**
Force field transform [3, 47]	kNN	74,93	66,67

V tabeli 6.3 je prikazana primerjava najnovejših metod za identifikacijo na podlagi uhljev z najboljšimi metodami, ki smo jih testirali mi: metoda rotacijsko invariantne kvantizacije lokalne faze in fuzija te metode z metodo normiranih slik. Metodi nista boljši od obstoječih, sta pa primerljivi. Fuzija R-N dosega natančnost $92,7\%$ in $87,64\%$ na bazah IIT Delhi I P in II P, metoda rotacijsko invariantne kvantizacije lokalne faze pa $79,82\%$ in $72,95\%$.

V primerjavi v tabeli 6.3 so prikazane le tiste metode, katerih avtorji so uporabili iste podatkovne baze uhljev kot mi – to sta poprocesirani bazi IIT Delhi I P in IIT Delhi II P in hkrati dosegajo dobre rezultate. Obstaja še veliko metod, katerih rezultati niso podani, ker so slabši in niso primerljivi s prikazanimi v tabeli 6.3.

Poglavje 7

Zaključek

V skladu s pričakovanji so rezultati pokazali, da se praviloma vse testirane metode na naši podatkovni bazi obnašajo slabše kot na obstoječih, a še vseeno primerljivo. Glavni razlog je v tem, da baza izvira iz nenadzorovanega okolja. Naša baza tako predstavlja večji izziv od obstoječih baz uhljev, s čimer se kaže njena uporabnost, kar rezultati tudi potrjujejo.

Razvito orodje za razpoznavo uhljev predstavlja korak standardizacije na področju primerjave metod za razpoznavo oseb na podlagi uhljev. Med nastajanjem tega dela je razvito orodje bistveno olajšalo testiranje in razvoj metod.

Z metodo rotacijsko invariantne kvantizacije lokalne faze smo pokazali, da je na določenih bazah boljša od metode histograma orientiranih gradientov, sploh v kombinaciji s fuzijo in primerljiva z najnovejšimi, ni pa boljša od danes najboljših. Razlogi, da so rezultati v splošnem nekoliko slabši, so tudi v neoptimiziranih odločitvenih modelih, ki pa niso bili fokus tega dela. Natančnost bi se dalo izboljšati tudi z detekcijo oblike uhlja in apliciranjem metod za pridobivanje značilk uhlja le znotraj detektirane krivulje [3].

Možnosti za izboljšave je še veliko, ena izmed njih je večja posvetitev odločitvenim modelom in optimizacija le teh. Dobro bi bilo tudi uporabiti konvolucijske nevronske mreže za odločitveni model. Zaradi močne paralelnosti jih je mogoče poganjati na grafičnih procesorjih, kar proces učenja pohitri tudi za faktor 40 [48] – to pospeši in olajša proces razvoja novih metod za razpoznavo oseb na podlagi uhljev.

V izdelavi je nova različica podatkovne baze uhljev z večjim številom oseb

in manjšim številom slik na osebo. Takšna baza bo, po pričakovanjih, težja od obstoječe in od katere druge trenutno dostopne.

V pripravi je tudi spletna stran z možnostjo nalaganja rezultatov, shranjenih značilk in naučenih odločitvenih modelov. Tako bomo olajšali sodelovanje med raziskovalci ter postopke primerjanja različnih pristopov. To lahko še dodatno pospeši razvoj na področju razpoznave oseb na podlagi uhljev.

Literatura

[1] A. Pflug and C. Busch, "Ear biometrics: a survey of detection, feature extraction and recognition methods," *Biometrics, IET*, vol. 1, no. 2, pp. 114–129, junij 2012.

[2] D. Frejlichowski and N. Tyszkiewicz, "The West Pomeranian University of Technology Ear Database – A Tool for Testing Biometric Algorithms," in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6112, pp. 227–234. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13775-4_23

[3] A. Kumar and C. Wu, "Automated human identification using ear imaging," *Pattern Recogn.*, vol. 45, no. 3, pp. 956–968, marec 2012. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2011.06.005

[4] P. Yan and K. Bowyer, "Empirical evaluation of advanced ear biometrics," in *Computer Vision and Pattern Recognition – Workshops (CVPR), IEEE Computer Society Conference on*, junij 2005, pp. 41–41.

[5] R. Raposo, E. Hoyle, A. Peixinho, and H. Proenca, "UBEAR: A dataset of ear images captured on-the-move in uncontrolled conditions," in *Computational Intelligence in Biometrics and Identity Management (CIBIM), IEEE Workshop on*, april 2011, pp. 84–90.

[6] Ž. Emeršič and P. Peer, "Ear biometric database in the wild," in *Bioinspired Intelligence (IWOBI), IEEE 4th International Work Conference on*, junij 2015, pp. 27–32.

[7] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 4, pp. 754–766, april 2011.

[8] A. Google, "Custom Search — Google Developers," 2015. [Online]. Available: https://developers.google.com/custom-search/

[9] Microsoft, "Bing Search API — Microsoft Azure Marketplace," 2015. [Online]. Available: http://datamarket.azure.com/dataset/bing/search

[10] Ž. Emeršič and P. Peer, "Toolbox for ear biometric recognition evaluation," in *IEEE International Conference on Computer as a Tool (EUROCON)*, 2015, pp. 716–721.

[11] R. Schultz and R. Ives, "Biometric data acquisition using Matlab GUIs," in *Frontiers in Education, FIE, 35th Annual Conference*, oktober 2005, pp. S1G–1.

[12] I. R. Institute, "Biometrics evaluation and testing (BEAT)," 2015. [Online]. Available: https://www.beat-eu.org/

[13] V. Štruc, "Inface: A toolbox for illumination invariant face recognition," 2009. [Online]. Available: http://luks.fe.uni-lj.si/sl/osebje/vitomir/face_tools/INFace

[14] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett, "The computer expression recognition toolbox (CERT)," in *Automatic Face and Gesture Recognition and Workshops (FG), IEEE International Conference on*, marec 2011, pp. 298–305.

[15] J. Lacirignola, P. Pomianowski, D. Ricke, D. Strom, and E. Wack, "Multimodal biometric collection and evaluation architecture," in *Homeland Security (HST), IEEE Conference on Technologies for*, november 2012, pp. 61–66.

[16] D. Gorodnichy, "Evolution and evaluation of biometric systems," in *Computational Intelligence for Security and Defense Applications (CISDA), IEEE Symposium on*, julij 2009, pp. 1–8.

[17] L. Omelina and M. Oravec, "Universal biometric evaluation system: Framework for testing evaluation and comparison of biometric methods," in *Systems, Signals and Image Processing (IWSSIP), 18th International Conference on*, junij 2011, pp. 1–4.

[18] B. Fernandez-Saavedra, R. Alonso-Moreno, J. Uriarte-Antonio, and R. Sanchez-Reillo, "Evaluation methodology for analyzing usability factors in biometrics," in *Security Technology, 43rd Annual International Carnahan Conference on*, oktober 2009, pp. 347–354.

[19] IBM, "Ibm watson," 2015. [Online]. Available: http://www.ibm.com/smarterplanet/us/en/ibmwatson/

[20] P. Grother and E. Tabassi, "Performance of biometric quality measures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 4, pp. 531–543, april 2007.

[21] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, junij 2012, pp. 3642–3649.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on*, vol. 1, junij 2005, pp. 886–893 vol. 1.

[23] N. Damer and B. Fuhrer, "Ear recognition using multi-scale histogram of oriented gradients," in *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), International Conference on*, julij 2012, pp. 21–24.

[24] D. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, IEEE International Conference on*, vol. 2, 1999, pp. 1150–1157.

[25] ——, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. [Online]. Available: http://dx.doi.org/10.1023/B%3AVISI.0000029664.99615.94

[26] J. Zhou, S. Cadavid, and M. Abdel-Mottaleb, "Exploiting color sift features for 2d ear recognition," in *Image Processing (ICIP), 18th IEEE International Conference on*, september 2011, pp. 553–556.

[27] J. Bustard and M. Nixon, "Robust 2d ear registration and recognition based on sift point matching," in *Biometrics: Theory, Applications and Systems, 2nd IEEE International Conference on*, september 2008, pp. 1–6.

[28] D. Kisku, H. Mehrotra, P. Gupta, and J. Sing, "Sift-based ear recognition by fusion of detected keypoints from color similarity slice regions," in *Advances in Computational Tools for Engineering Applications (ACTEA), International Conference on*, julij 2009, pp. 380–385.

[29] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451 – 461, 2003, biometrics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320302000602

[30] T. Santhanam and M. Padmavathi, "Comparison of k-means clustering and statistical outliers in reducing medical datasets," in *Science Engineering and Management Research (ICSEMR), International Conference on*, november 2014, pp. 1–6.

[31] VLFeat, "Vlfeat matlab," 2015. [Online]. Available: http://www.vlfeat.org/install-matlab.html

[32] A. Bosch, A. Zisserman, and X. Muoz, "Image classification using random forests and ferns," in *Computer Vision, (ICCV) IEEE 11th International Conference on*, oktober 2007, pp. 1–8.

[33] B. P. Arun, K. Roy, A. Thoufeeq, V. Mohanraj, V. Vaidehi, and K. Ranajit, "Facial feature extraction and recognition using Deformable Parts Model and DSIFT," in *Signal Processing, Communication and Networking (ICSCN), 3rd International Conference on*, marec 2015, pp. 1–6.

[34] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761 – 767, 2004, british Machine Vision Computing 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885604000435

[35] V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," in *3rd International Conference on Image and Signal Processing*, ser. ICISP '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 236–243. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-69905-7_27

[36] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila, "Recognition of blurred faces using local phase quantization," in *Pattern Recognition (ICPR), 19th International Conference on*, december 2008, pp. 1–4.

[37] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, julij 2002.

[38] V. Ojansivu, E. Rahtu, and J. Heikkilä, "Rotation invariant local phase quantization for blur insensitive texture analysis," in *Pattern Recognition (ICPR), 19th International Conference on*, december 2008, pp. 1–4.

[39] A. Ross and A. Jain, "Multimodal biometrics: An overview," in *Signal Processing Conference, 12th European*, september 2004, pp. 1221–1224.

[40] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, september 1995. [Online]. Available: http://dx.doi.org/10.1023/A:1022627411411

[41] J. Rodriguez, A. Perez, and J. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 3, pp. 569–575, marec 2010.

[42] Mamta and M. Hanmandlu, "Robust ear based authentication using local principal independent components," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6478–6490, november 2013. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2013.05.020

[43] H. A. Benzaoui, A. and A. Boukrouche, "Sensitivity analysis of k-fold cross validation in prediction error estimation," *Journal of Electronic Imaging*, vol. 23, no. 5, pp. 569–575, september 2014.

[44] J. Kannala and E. Rahtu, "Bsif: Binarized statistical image features," in *Pattern Recognition (ICPR), 21st International Conference on*, november 2012, pp. 1363–1366.

[45] A. Basit and M. Shoaib, "A human ear recognition method using nonlinear curvelet feature subspace," *Int. J. Comput. Math.*, vol. 91, no. 3, pp. 616–624, marec 2014. [Online]. Available: http://dx.doi.org/10.1080/00207160.2013.800194

[46] T.-S. Chan and A. Kumar, "Reliable ear identification using 2-d quadrature filters," *Pattern Recogn. Lett.*, vol. 33, no. 14, pp. 1870–1881, oktober 2012. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2011.11.013

[47] D. Hurley, B. Arbab-Zavar, and M. Nixon, "The ear as a biometric," in *Signal Processing Conference, 15th European*, september 2007, pp. 25–29.

[48] Intelligenceagent, "intelligenceagent / cudacnn-public – bitbucket," 2015. [Online]. Available: https://bitbucket.org/intelligenceagent/cudacnn-public